

机器人竞技系列



自律型机器人制作

〔日〕西山一郎 兆十 著



AUTONOMOUS ROBOT



科学出版社

OHM社







(TP-1700.0101)

责任编辑 赵丽艳 樊友民

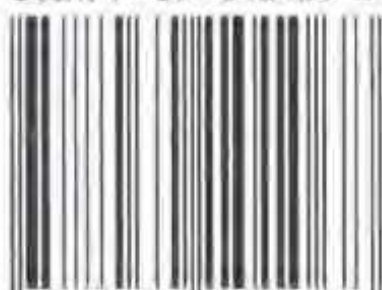
责任制作 魏 谨

封面制作 李 力



-  机器人 C 语言机电一体化接口
-  有视觉机器人制作
-  机器人竞赛指南
-  机器人制作宝典
-  机器人组装大全
-  **自律型机器人制作**

ISBN 7-03-010106-5



9 787030 101068 >

ISBN 7-03-010106-5/TP · 1700

定价: 25.00 元

机器人竞技系列



自律型机器人制作

〔日〕西山一郎 兆 十 著
耿连发 潘维林 译
孙昌志 校



科学出版社 OHM 社

2002

144-63/04

图字:01-2001-0410 号

Original Japanese edition

Robobooks Jiritsugata Robot Seisaku Bible

by Ichirou Nishiyama and Megaten

Copyright © 2000 by Ichirou Nishiyama and Megaten

Published by Ohmsha, Ltd.

This Chinese language edition is co-published by Ohmsha, Ltd. and Science Press

Copyright © 2002

All rights reserved

本书中文版版权为科学出版社和 OHM 社所共有

RoboBooks

自立型ロボット製作バイブル

西山一郎 メガテン オーム社 2000年 第1版第1刷

图书在版编目(CIP)数据

自律型机器人制作/(日)西山一郎,兆十著;耿连发,潘维林译. 北京:科学出版社,2001
(机器人竞技系列)

ISBN 7-03-010106-5

I. 自… II. ①西…②兆…③耿…④潘… III. 机器人-制作 IV. TP24.2

中国版本图书馆 CIP 数据核字(2002)第 008183 号

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科学出版社 OHM 社 出版

北京东黄城根北街 16 号 邮政编码:100717

<http://www.sciencep.com>

中国科学院印刷厂 印刷

科学出版社发行 各地新华书店经销

2002 年 3 月第 一 版 开本:R5(720×1000)

2002 年 3 月第一次印刷 印张:12 1/2

印数:1 5 000 字数:215 000

定 价: 25.00 元

(如有印装质量问题,我社负责调换〈新欣〉)

前 言

面临电子时代的到来,计算机已经深入到人们生活的各个领域。可以毫不夸张地说,不会使用计算机的人将来连生存都成问题。因此,如何利用好计算机就变得非常重要了。

近几年,机器人竞赛非常盛行,以至于许多大专院校都开始利用机器人竞赛活动来促进计算机教育及培养学生动手制作的能力。这种教育使至今被许多人认为是可望而不可及的计算机世界走进我们的生活,同时,对培养计算机时代所需要的人才起到了非常重要的作用。

作为本书的宗旨,首先是要使任何人都能够制作出自己的机器人。同时,还希望藉此为即将到来的电子信息时代培养出一批充满自信的生力军。

通过制作机器人可以获得如下感受:

- 自己制作的机器人能够按照设计要求完成动作时的喜悦;
- 从策划、设计到制作、测试等一系列的开发过程均由自己亲手完成的成就感;
- 体验参加机器人竞赛时的紧张与激动的心情;
- 可以向具有无限开发潜力的机器人领域进行永无止境的挑战。

如果能通过制作机器人而体会到上述乐趣,哪怕只是其中的一种,就足以说明通过制作机器人进行教育的目的已经达到了。

本书在编写过程中非常注意的一点,就是要使那些不具备电子学知识的读者也可以使用计算机轻松地完成机器人的制作。另外,现在已经进入了网络时代,为了那些具备上网条件的读者,我们还在书后附上了 CD-ROM¹⁾,提供了可以获得大量相关信息的网站地址,以便读者自由查询。

著 者

1) 中文版暂不附 CD-ROM。——译者注

目 录

Chapter 1 自律型机器人竞赛

1.1	自律型机器人竞赛	2
1.2	相扑机器人大会	3
1.3	微型鼠竞赛大会	5
1.4	日本微型计算机圣杯机器人竞赛	8
1.5	三叉戟机器人竞赛大会	9

Chapter 2 基本的电子部件

2.1	电阻和电容	12
2.2	半导体	15
2.3	逻辑电路	18
2.4	运算放大器	20
2.5	机器人设计工具简介	22
2.6	电路设计、基板设计 CAD	22

3.1	计算机	26
3.2	H8-3048F 单片机	27
3.3	集成定时单元(ITU)	30
3.4	A/D 转换器、D/A 转换器	35
3.5	中 断	38
3.6	充分利用串行通信 R232C 端口	42

4.1	制作 H8 微型计算机插件板	48
4.2	市售微型计算机插件板的选定	53
4.3	AKI-H8 用 I/O 插件板的制作	56
4.4	制作快速存储器写入器插件板	56

5.1	必要的开发环境	62
5.2	C 语言开发环境的准备	63
5.3	使用 GCC 编制 H8 微型计算机程序	63

Chapter 6 机器人的嘴

6.1	在 LED 上表示	76
6.2	全色 LED	77
6.3	让压电蜂鸣器或扬声器发出声音	78
6.4	液晶显示器(LCD)	79
6.5	由扬声器发出声音	81
6.6	在个人电脑上通过通信端口输出文字和数字 ...	82
6.7	用无线遥控器驱动机器人奔跑	84

Chapter 7 机器人的五感

7.1	按钮开关	86
7.2	数字开关	87
7.3	接触传感器	88
7.4	由电位器模拟输入	89
7.5	光传感器	89
7.6	超声波传感器	98
7.7	旋转编码器(光电编码器)	101
7.8	回转仪传感器	104
7.9	G 传感器	105
7.10	对人反应的传感器	106
7.11	感压传感器	107

7.12	PSD 距离传感器	108
7.13	在 AD 转换器上输入声音	110

Chapter 8 机器人的手和脚

8.1	电动机	114
8.2	电动机驱动器	119
8.3	电动机的控制	126
8.4	机器人臂结构和控制	128

Chapter 9 机器人的能量

9.1	蓄电池	134
9.2	充、放电器	137
9.3	CPU 用的 5V 电源电路	137

Chapter 10 相扑机器人的制作方法

10.1	机器人的构成	140
------	--------------	-----

10.2	相扑机器人的车身	140
10.3	车板(电话卡)的安装方法	142
10.4	吸引的方法	144
10.5	相扑机器人臂的结构	151
10.6	机器人的制作实例	155
10.7	高速行走的机器人	166
10.8	吸引型机器人的制作实例	167

Chapter

11

直线追踪机器人的制作方法

11.1	车身的制作方法	172
11.2	驱动电动机插件板	174
11.3	光传感器插件板	175
11.4	安装长枪部件	175
11.5	机器人的程序实例	176

Chapter

12

制作小型机器人

12.1	小型机器人概述	180
12.2	小型机器人的机械结构	181
12.3	小型机器人的回路	185
12.4	小型机器人动作实例	185
12.5	小型机器人的实例程序	186

chapter

1

自律型机器人竞赛

本章介绍日本国内进行的自律型机器人竞赛中的高能动性机器人。在自律型机器人竞赛中，参赛机器人可以分为竞赛型（如相扑机器人）和争速度型（如线路追踪型机器人）两类。本章将对这两种机器人进行介绍。希望参加机器人竞赛的读者在掌握本章内容的基础上，进入后面章节的学习。

1.1 自律型机器人竞赛

所谓自立,对于人来讲,一般是指具有独立生活能力的成年人。然而,这里所说的“自立”,则是指婴儿从会爬逐渐成长,刚刚能够站立时的样子。

本书中所介绍的“自立型机器人”,指的就是这样的机器人。大学中的控制专家们称这种机器人为“自律型机器人”,考虑到“自律”一词在日文中理解上的问题,故用“自立”——自己站立,来命名这种机器人¹⁾,这样一来即使是初学者,也不会因名词而产生畏难的心理。

即使说自己站立,目前也只有本田(Honda)的机器人刚刚开始能用两脚步行,机器人竞赛中还未出现过。也许不久的将来,这样的机器人能在机器人竞赛场登台竞技,而现在的机器人多数装载微型计算机,由装在微型计算机上的程序判断连接在微型计算机上的传感器的状态,再用电动机和轮胎使机器人像汽车那样运动。

自律型机器人的制作网址:

本田的网址:

<http://www.honda.co.jp/robot/>

索尼 AIBO 的网址:

<http://www.jp.albo.com/>

自律型机器人竞赛 20 年前就开始举办了。最近机器人竞赛已经扩大到世界范围,形成了所谓召开“机器人奥林匹克大会”的状况。

机器人创作国际竞赛大会(机器人节日)

所谓机器人创作国际竞赛大会将是 2001 年第一次举办的机器人技术和机器人工作者的盛典,当然可称之为机器人奥林匹克大会。

第一次在 2001 年举办,之后计划隔几年举办一次。2001 年举办地点是夏季大会在以日本的大阪府为中心的关西地区,秋季大会在神奈川县。到 2000 年 9 月 26 日,已经是举办的第 24 次机器人竞赛了。

机器人创作国际竞赛大会的网址:

<http://www.robofesta.net/>

现在进行的机器人大会的网址:

机器人奖杯赛场

<http://www.roboocup.org/02.html>

机器人大奖赛场

<http://www.etl.go.jp/etl/amss/RobotGroundPrix/>

1) 因“自律型机器人”在我国已是一个人们熟知的专业术语,所以本书翻译中均将原书中的“自立型机器人”译为“自律型机器人”。 译者注

智能机器人竞赛赛场

<http://www.robotics.is.tohoku.ac.jp/Inrof-J.html>

全日本相扑机器人大会赛场

<http://www.fsi.co.jp/sumo/>

微型鼠赛场

<http://www.bekkoame.ne.jp/~ntf/mouse/mouse.html>

川崎机器人赛场

<http://www.kawasaki-net.ne.jp/topix/robo/robo00/index.htm>

微型机械装置竞赛赛场

<http://www.cc.toin.ac.jp/sc/hayashi-lab/top.html>

日本微型计算机圣杯赛场

<http://www.mcr.gr.jp/index2.html>

富士电视机器人圆形比赛赛场

<http://www.fujitv.co.jp/jp/robot/index2.html>

与本书有关公司的网址:**欧姆社机器人控制杂志**

<http://www.ohmsha.co.jp/robocon/index.htm>

最佳技术株式会社

<http://www.besttechnology.co.jp/>

1.2 相扑机器人大会

相扑机器人大会由富士软件 ABC 股份有限公司主办,竞赛方法实行淘汰制。

第一次相扑机器人大会是在 1989 年召开的。从 1993 年开始各地区相继召开相扑机器人大会,由各地区大会进入决赛的选手参加决赛大会。与此同时,也召开了高中生的相扑机器人大会(参见照片 1.1)。参加多届大会的人数也在大幅度地增加。

在相扑机器人中有自律型部分和无线方向遥控型部分,相扑机器人在直径为 154cm 的圆形摔跤场中竞赛。机器人大小为 200 mm×200mm 以下,重量为 3kg 以下,高度不限制。

对于自律型机器人,在裁判员下达“开始”命令后,按机器人起动按钮,5 秒钟倒计时后(这期间机器人不动),机器人本身确认对手的位置,然后出击。换句话说,按下按钮后,操作者在摔跤场边线外,只能一动不动地定睛注视自己机器人的动作(参见照片 1.2)。

基本的规则只有这些,其余的规则都是为了进行公平、公正的竞赛而设立的。对于制作“真正的”机器人的各位读者来说,这些都是没有问题的内容。

摔跤场的说明如图 1.1 所示,为了方便机器人检测,四周使用白色线。



• 照片 1.1 相扑机器人竞赛高中生大会赛场一角



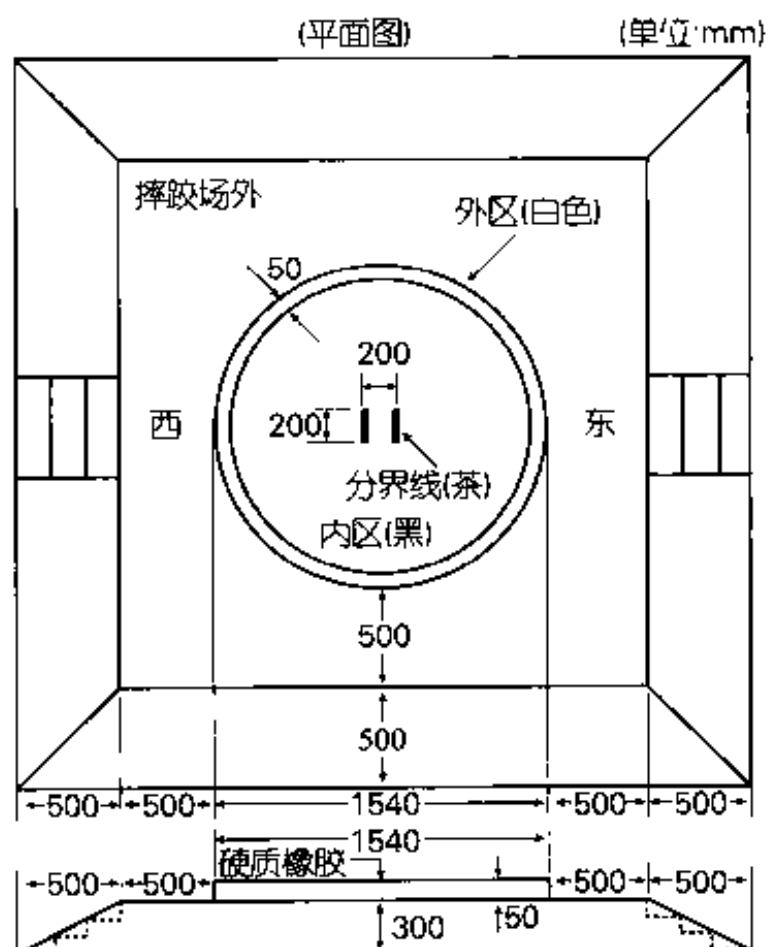
• 照片 1.2 竞赛开始

1.2.1 全日本相扑机器人大会竞赛规则

相扑机器人竞赛规则在富士软件 ABC 股份有限公司的网页上已经登载了,所以这里仅做概要说明,大会参加者事前必需熟读这些规则。

富士软件 ABC(株)全日本相扑机器人大会网址:

<http://www.fsf.co.jp/sumo/>



- 直径 1.54m
- 材质 硬质橡胶
- 色
- 摔跤场内区 黑
- 分界线 茶
- 外区 白
- 摔跤场的照度 < 1000lx

• 图 1.1 相扑机器人摔跤场的概略图

1.2.2 机器人规格

机器人要装在外形宽 20cm、深 20cm、高度任意的四角形的框架内。包括附属部件重量为 3000g 以下。

1.2.3 摔跤场场内的定义

所谓摔跤场场内是指内区(黑色)

部分和外区(白色)部分,其它部分称为摔跤场场外。摔跤场为高度 5cm、直径 154cm 的铝板,在其上面贴上黑色硬质橡胶作成圆形(参见图 1.1)。分界线在中心点两侧 20cm 的位置,用宽 2cm,长 20cm 的茶色线表示。摔跤场内、外的分区线为 5cm 宽的白色线(白色线以内为摔跤场场内)。

1.3 微型鼠竞赛大会

微型鼠竞赛大会自 1977 年 5 月 IEEE 的《光谱》(Spectrum)杂志上载文提倡以来,在世界各国相继开展起来。

IEEE 的网址:

<http://www.ieee.org/>

日本在 1980 年举办第一次微型鼠竞赛大会。每年有 100 台左右的微型鼠,以它们的头脑和速度进行竞赛。和其它机器人竞赛相比,它是最有实绩的理智竞赛。在日本微型鼠大会上进行以下项目:

1. 微型鼠竞赛,
2. 机器人竞赛,
3. 微型赛艇竞赛。

微型鼠协会的网址:

<http://www.bekkoame.ne.jp/~ntf/mouse/mouse.html>

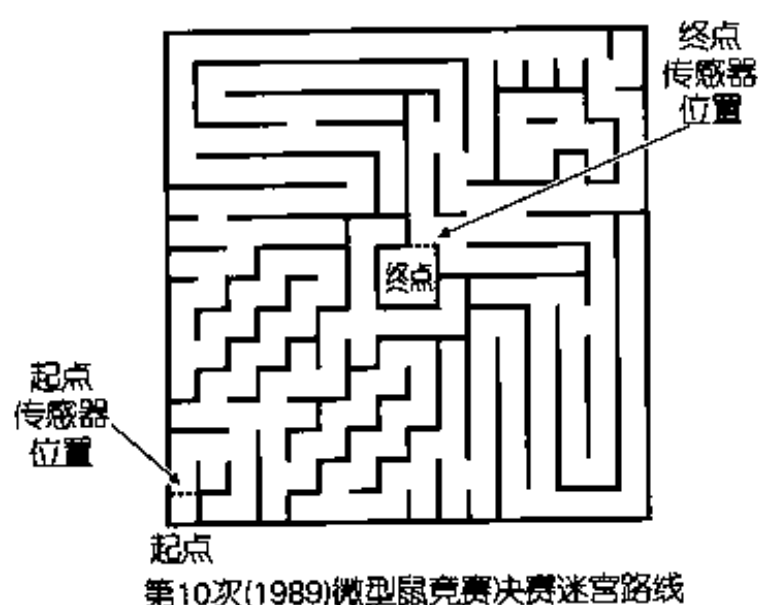
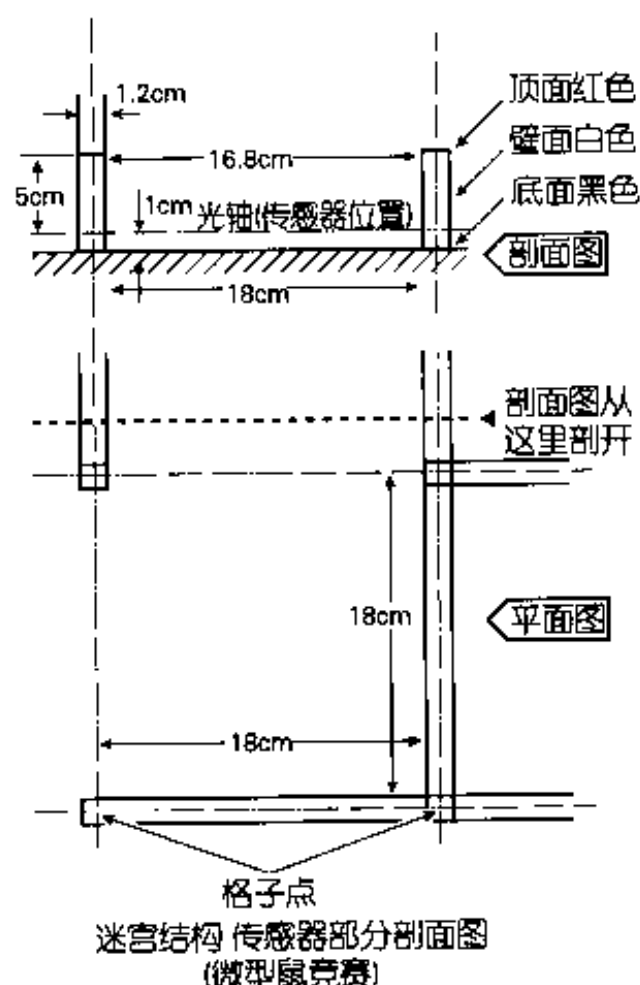
1.3.1 微型鼠竞赛

微型鼠竞赛如照片 1.3 所示,项目为迷宫探路,从起点到终点以最短的路线、最短的时间到达者为优胜。



• 照片 1.3 微型鼠竞赛大会的会场

迷宫由 16×16 个区组成, 起点设在拐角处, 终点设在中央, 占 4 个区。每 1 个区为 $180 \text{ mm} \times 180 \text{ mm}$ 大小, 间壁高为 50 mm , 厚度为 12 mm 。顶面涂红色、侧面涂白色, 底面涂黑色(参见图 1.2)。

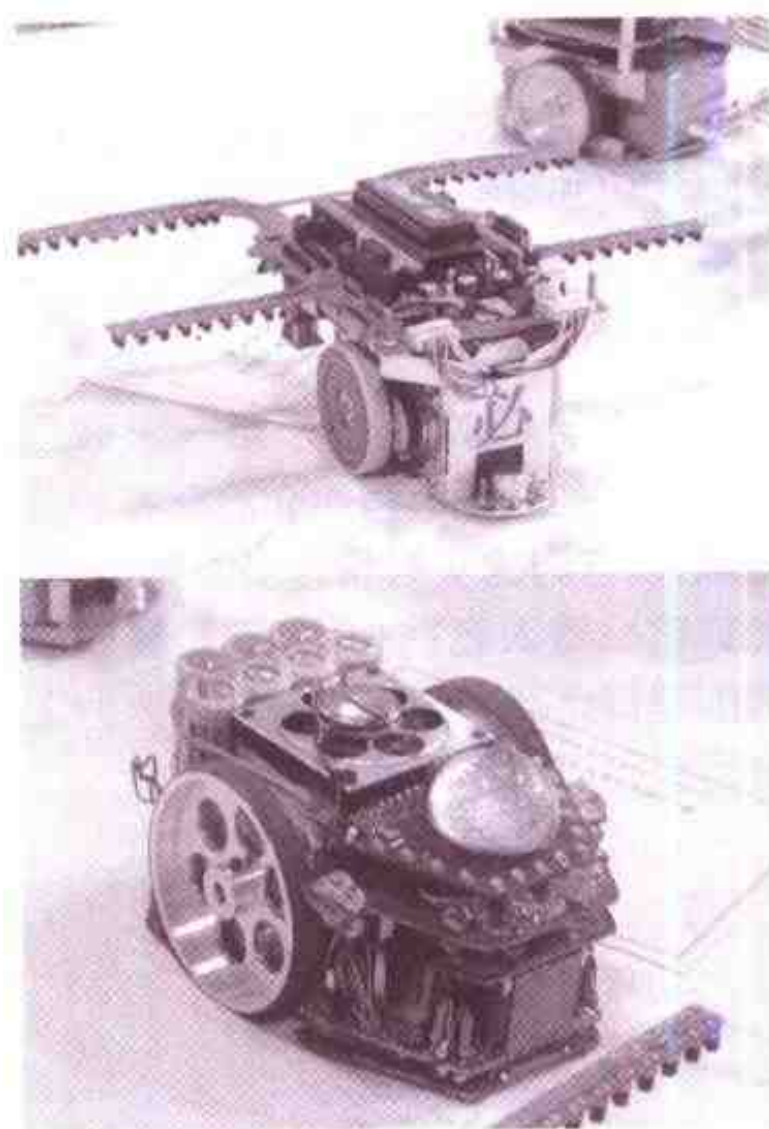


• 图 1.2 微型鼠竞赛路线

机器人的大小为长、宽各不超过 250 mm 。高度和重量没有限制。把从

起点到终点用最短时间行走的记录作为这个机器人的记录。另外, 机器人必须是自律型, 行走过程中不能用手接触机器人。不言而喻, 用无线遥控更是不可以的, 不允许事前将迷宫的信息通知微型鼠。而且, 竞赛过程中禁止变更重量或输入、存放程序(指传送给微机存储器)。

当微型鼠开始行走的时候, 不能给它迷宫信息。因此, 机器人为了获得迷宫路线信息, 要由相当于眼睛的传感器部分感知有没有墙壁, 用相当于脚的驱动部分移动, 以及用相当于大脑的计算机部分认路。机器人就是由这几部分构成的。照片 1.4 示出了两种典型的机器人。



• 照片 1.4 微型鼠机器人

微型鼠竞赛程序是高级程序,所以在本书中没有写入,而本书中采纳的技术和读者具有的软件技术,也可能是向这种竞赛的挑战。

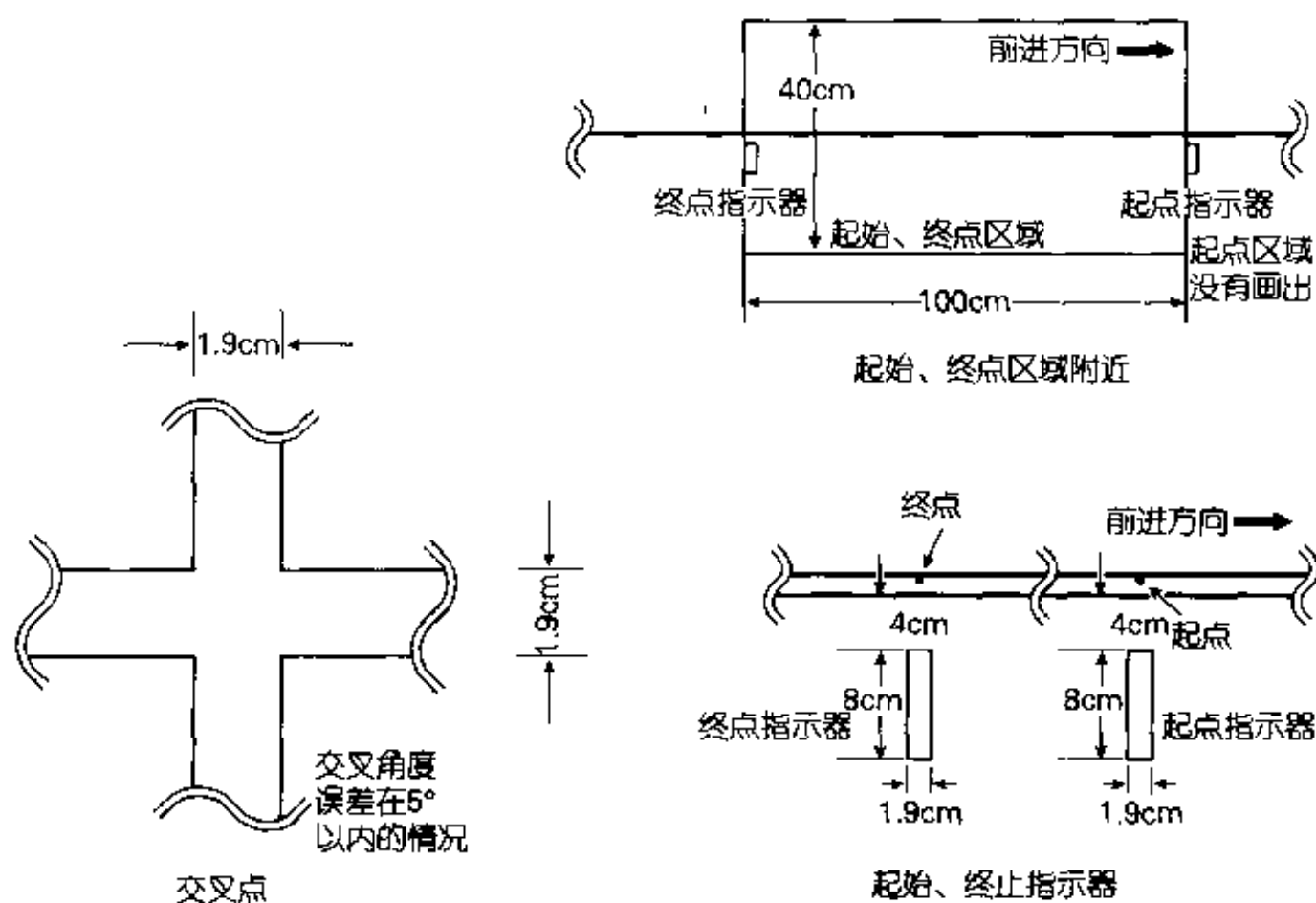
1.3.2 机器人竞赛

机器人竞赛是由第7届全日本微型鼠大会举办的。机器人沿着跑道移动,进行速度比赛,所以根据跑道指示器,可以判断直线、曲线和起动位置

(参见图 1.3)。第 13 次全日本微型鼠大会不仅改变了跑道路线,参加赛跑的机器人台数也增加了,是一次有趣的竞赛。

在 5 分钟内,允许行走 5 次,所以必须采取以下策略:第 1 次行走判断直线、曲线,第 2 次以后的行走进行加速、减速以及可能高速行走等等。

机器人大小和微型鼠相同,是 250mm × 250mm。但是高度限制在 200mm 以下。

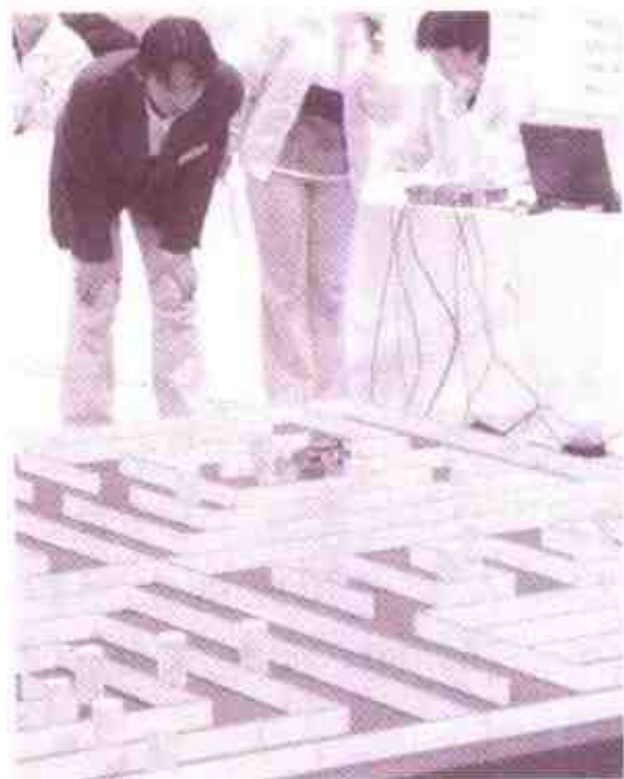


• 图 1.3 机器人竞赛跑道(路线)

1.3.3 微型赛艇竞赛

微型赛艇竞赛是从第 13 届全日本微型鼠大会开始的。在迷宫的路上事先放置圆筒形罐子,将其放倒即可得分,一定时间内,放倒罐子多的机器

人获胜(参见照片 1.5 所示)。在微型鼠上安装相当于手这样的东西,就成为比微型鼠更高级的机器人。我们预想随着具有罐子探索技术、放倒等新技术的机器人的不断出现,今后的机器人竞赛将是更有欣赏价值的竞赛。



• 照片 1.5 抓住圆罐的机器人

1.4 日本微型计算机圣杯机器人竞赛

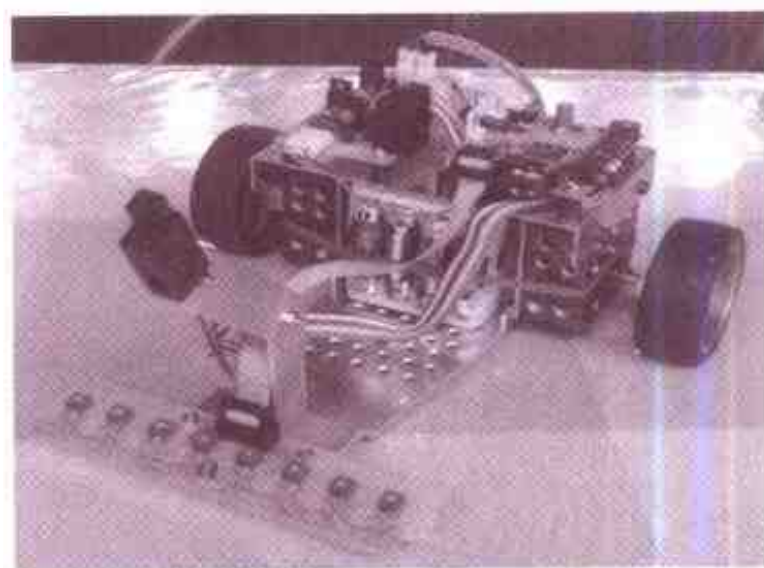
8

日立微型计算机圣杯机器人竞赛始于 1995 年,现在叫做日本微型计算机圣杯机器人竞赛。本书介绍的机器人竞赛大会,限定使用日立制 H8CUP。参加大会的有高中生、大学生以及一般社会人士,他们参与各自熟悉项目的竞赛。

通过“手工制作”推进教育,培养对新技术的关心和向往,促进机电一体化技术的发展与提高,是大会的宗旨。

日本微型计算机圣杯的网址:
<http://www.mcr.gr.jp/index2.html>

1.4.1 机器人规格



• 照片 1.6 微型计算机车

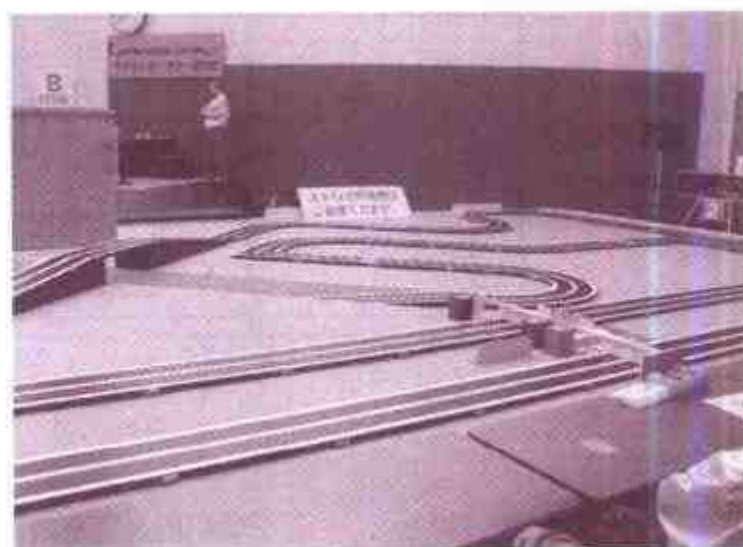
- ① 搭载 H8 微型计算机的完全自律型机器人,是竞赛中接触跑道表面行走的机器人;

- ② 电源或能源为 5 号碱性电池,8 只以内;
- ③ 机器人外形宽 300mm、高 200mm 以内,对形状、长度、重量及材质没有限制。照片 1.6 示出代表性的微型计算机车。

1.4.2 跑道规格

跑道宽 300mm,以黑色的消光丙烯酸塑料薄膜制成,中心线由 20mm 宽的白线及其两侧各宽 10mm 的灰色线表示,路侧线可用宽 20mm 的白线。跑道整体由直线、曲线、弯线、S 形线,以及上、下坡道构成,跑道路线图在当

日大会上公布。照片 1.7 所示为第 6 届大会的跑道(路线)。



• 照片 1.7 日本微型计算机圣杯机器人竞赛会场

1.5 三叉戟机器人竞赛大会

三叉戟机器人竞赛在机械学会主办的机器人锦标赛中进行。

搭载了长枪的微型计算机的自律型机器人,沿着圆周形的跑道旋转,以攻击到在跑道线上的左右标的为得分(参见图 1.4)。机器人在 100 秒以内,攻击在圆周形跑道线上左右的各种类型和大小的标的,看谁攻击正确,并且能够多攻击为取胜。照片 1.8 是三叉戟机器人竞赛会场。

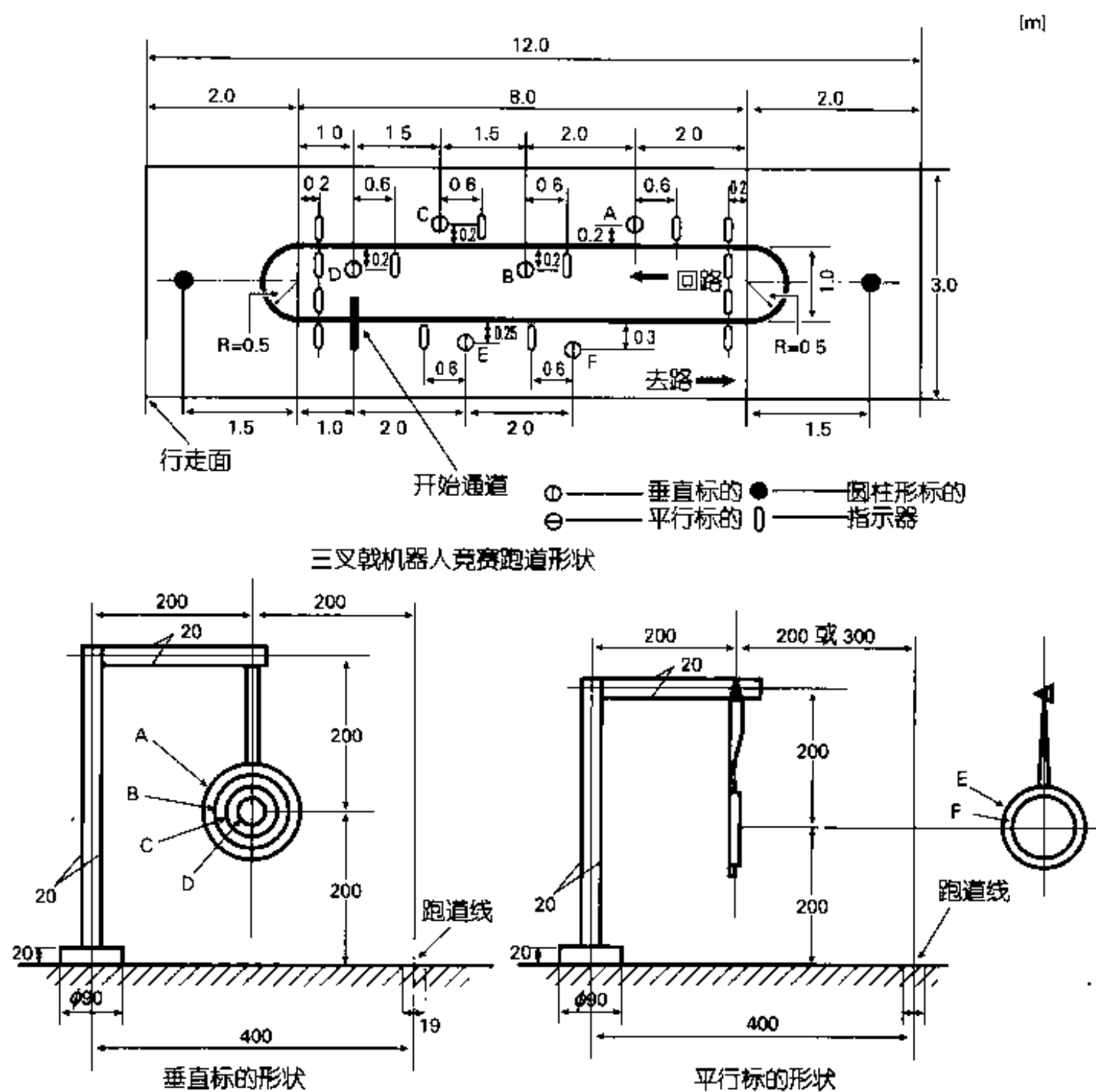
1.5.1 三叉戟机器人的代表实例

照片 1.9 表示典型的三叉戟机器人。获冠军的是步进电机型机器人,

亚军是直流电动机型机器人。今后预测机器人将向直流电动机方向发展,本书介绍制作直流电动机型机器人。



• 照片 1.8 三叉戟机器人竞赛会场



• 图 1.4 三叉戟机器人的竞赛跑道



冠军机器人



亚军机器人

• 照片 1.9 典型的三叉戟机器人

chapter

2

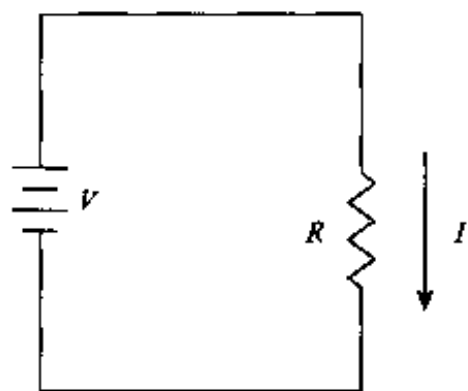
基本的电子部件

本章介绍制作参加机器人竞赛的机器人所必需的基本电子部件及电子电路。通过对其原理及思维方法的理解，可以提高开发机器人的效率。在制作的机器人出现故障时，往往很烦恼，这时，应冷静下来，从最基本的问题开始考虑。

2.1 电阻和电容

2.1.1 电阻

电阻有阻碍电流流动的性质,所以在限制电流或设定电压时使用它,电阻值表示阻碍电流流动的能力的大小,单位为欧[姆], 1000Ω 可以写成 $1k\Omega$, $1000k\Omega$ 可以写成 $1M\Omega$ 。电阻的符号如右上图所示。如图 2.1 那样,在电阻两端施加电压时,我们认为其电压(V)、电流(I)和电阻(R)有如图 2.1 所示的关系。其中电池组的符号如右图所示。



• 图 2.1 电压、电流和电阻的关系

图 2.1 给出基本的电路。这时下式成立(称为欧姆定律):

$$V = I \times R$$

其次,我们考虑一下电阻的连接。电阻串联连接时,总电阻为两个串联电阻之和:

$$R = R_1 + R_2$$

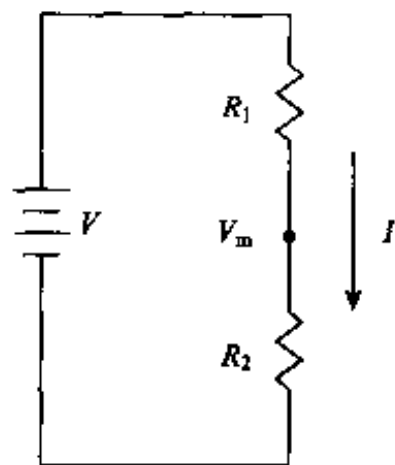
另外,若将相同电阻值的两个电阻并联连接时,其总电阻为其阻值的一半:

$$1/R = 1/R_1 + 1/R_2 \text{ 或 } R = R_1 / 2$$

将两个电阻连接成如图 2.2 的情况,我们看一看其中点电压 V_m 。这时的电流 $I = V / (R_1 + R_2)$,因为中点电压为 $I \times R_2$,所以,

$$V_m = V \times R_2 / (R_1 + R_2)$$

这样,改变电阻值就能得到需要的电流值,根据组合电阻就可以得到需要的电压值。基于这种考虑可制成需要的电子电路。



• 图 2.2 连接 2 个电阻的电路

一般,我们使用的电阻形状如照片 2.1 所示,上面印刷 4 条彩色标记。这个彩色标记表示电阻值,可以参考表 2.1 的彩色表读出电阻值。例如, $1k\Omega$ 电阻上印刷成茶色、黑色、红色,最后的颜色表示电阻的精度(容差)。



• 照片 2.1 电 阻

表 2.1 彩色标记表

颜色	第1颜色带	第2颜色带	第3颜色带	第4颜色带
	第1数字	第2数字	乘方数	标称电阻值容差
黑	0	0	10^0	—
棕	1	1	10^1	$\pm 1\%$
红	2	2	10^2	$\pm 2\%$
橙黄	3	3	10^3	—
黄	4	4	10^4	—
绿	5	5	10^5	$\pm 0.5\%$
蓝	6	6	10^6	—
紫	7	7	10^7	—
灰	8	8	10^8	—
白	9	9	10^9	—
金	—	—	10^{-1}	$\pm 5\%$
银	—	—	10^{-2}	$\pm 10\%$
—	—	—	—	$\pm 20\%$

2.1.2 电 容

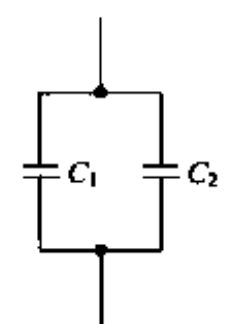
对于直流电,电容起“储蓄电”的作用,这时“电”不能流过电路的其他地方,而对交流,则能流过,并且电容的电阻值根据频率而变化。巧妙地利用电容这种特性,就能选定需要的频率或切断直流分量。表示电容容量的单位是法[拉](F),1 法[拉]等于 10^6 微法[拉]($1\mu\text{F} = 10^{-6}\text{F}$),1 微法[拉]等于 10^6 皮法[拉]($1\text{pF} = 10^{-6}\mu\text{F}$)。

电容符号,不必表示极性(+/-)的用 $\text{—}\text{||}\text{—}$ 表示,需表示极性的用 $\text{—}\text{||}\text{—}$ 表示。

电容串联连接(参见图 2.3)时, $C = 1/(1/C_1 + 1/C_2)$,并联连接(参见图 2.4)时为 $C = C_1 + C_2$ 。这个计算公式和计算电阻时的公式正相反。



• 图 2.3 电容串联连接



• 图 2.4 电容并联连接

在机器人中经常使用的电容有以下三种：

- ① 叠片式陶瓷电容器(参见照片 2.2): 主要用作旁路电容, 将电源上搭载的交流噪声旁路接地。



• 照片 2.2 叠片式陶瓷电容器

- ② 云母电容器(参见照片 2.3), 在高频电路中用在 CPU 时钟脉冲等部分。

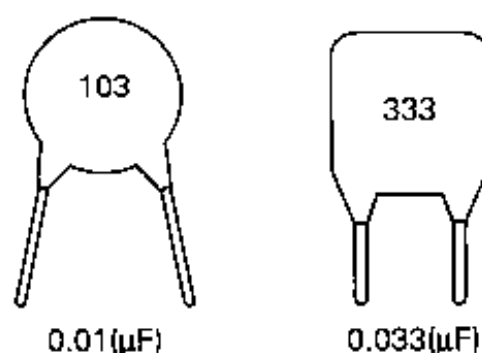


• 照片 2.3 云母电容器

上述 2 种电容器的容量表示读法如图 2.5 所示。

$$[103]: 10 \times 10^3 = 10000(\text{pF}) \\ = 0.01(\mu\text{F})$$

$$[333]: 33 \times 10^3 = 33000(\text{pF}) \\ = 0.033(\mu\text{F})$$



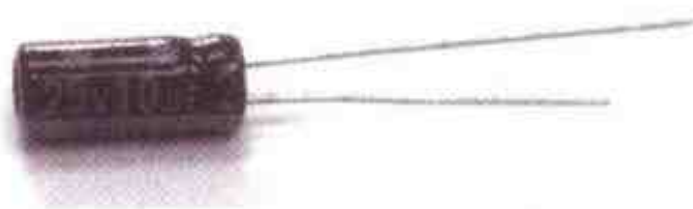
• 图 2.5 电容器容量表示

- ③ 铝电解电容器(参见照片 2.4): 铝电解电容器组装在电源电路中、起稳定电压的作用。



• 图 2.6 电解电容器的极性

电解电容器的容量及其两端的耐压程度(两极间能够加上的最大电压)同时表示。另外, 这种电容器有极性, 一般是引线长的一端为正极(参见图 2.6)。



• 照片 2.4 铝电解电容器

2.1.3 线圈

在线圈中直流分量容易通过, 交流分量难通过。这和电容的性质正相反。本书中, 没有使用线圈元件, 而在

举例的电动机中使用了线圈,在电动机中流过脉冲电流时,这个线圈的特性起作用。

将阻碍交流电流的性质称为电

感,对交流电是阻力,单位为亨[利](H)。这个值越大,交流电流越难通过,并且频率越高,电流也越难通过。

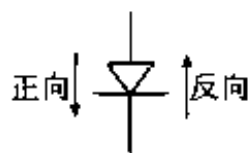
2.2 半导体

2.2.1 二极管

具有既能流通电流又不能流通电流的性质的物质称为半导体,而这种容易流通电流的性质取决于半导体中自由电子和空穴的多少。根据这两种载流子数量的不同,可以分成P型半导体和N型半导体两类。

P型半导体中空穴比自由电子多,通过从外部来的自由电子的作用使空穴依次移动下去,电流就可以流通。这时,具有负电荷性质的自由电子向具有正电荷性质的空穴移动过来,仿佛是正确的空穴在移动,将其称为P(空穴形)型半导体。相反,N型半导体中自由电子比空穴数多,电子移动形成电流。因自由电子带负电,所以称为N(电子形)型半导体。

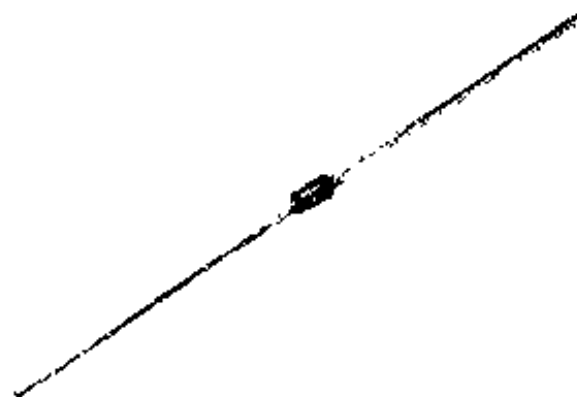
基本的PN结二极管就是具有空穴多的P型半导体与具有自由电子多的N型半导体形成的PN结。符号如右图所示。



当正向电流流通时,自由电子通过空穴移动,形成电流。施加反向电压时,由于电压差,空穴和自由电子完

全分离成两极,所以电流不能流通。

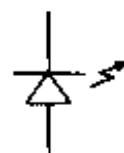
利用上述性质的就是二极管,加正向电压时电流流通,而加反向电压时电流不能流通。这称为整流作用,是二极管的最大特点。



• 照片 2.5 二极管

2.2.2 发光二极管

对于发光二极管(LED),当二极管中流过正向电流时,自由电子的能级降低,余下部分的能量以光的形式发出,这就是发光的原理。通过改变发出光的波长,可以制造出各种各样颜色的发光二极管。其符号如上图所示。



在发光二极管中发蓝色光的管子制造困难,而最近价格又降低了。另

外,也可把数个发光二极管并列起来组成发光二极管阵列(参见照片 2.6)。



• 照片 2.6 发光二极管(上)与发光二极管阵列(下)

还有,能发出红外光的发光二极管可以用在红外线传感器中的红外发光部件上。

2.2.3 光电二极管

光电二极管的基本结构和发光二极管相同,不同之处是当它遇到光时就有与光通量成正比的电流反方向流动,将这个反向流动的电流称为光电流。这种性质可以使用在传感器上。

光电二极管和发光二极管正相反,它是由电子获得光能量,变成高能

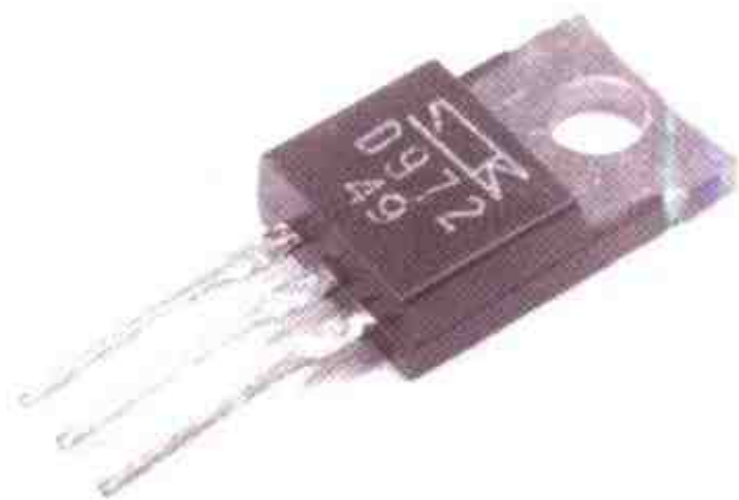
级状态。在这种状态下,电子容易运动,这就形成光电流,其符号如右图所示。其形状如照片 2.7 所示。



• 照片 2.7 光电二极管

2.2.4 晶体管

晶体管分为双极型晶体管和场效应型晶体管(FET)等等,它们被用在放大器或开关中。晶体管的外形如照片 2.8 所示。



• 照片 2.8 晶体管

晶体管如表 2.2 所示,大体分为双极型晶体管和场效应型晶体管(FET)。而且在双极型晶体管中又有

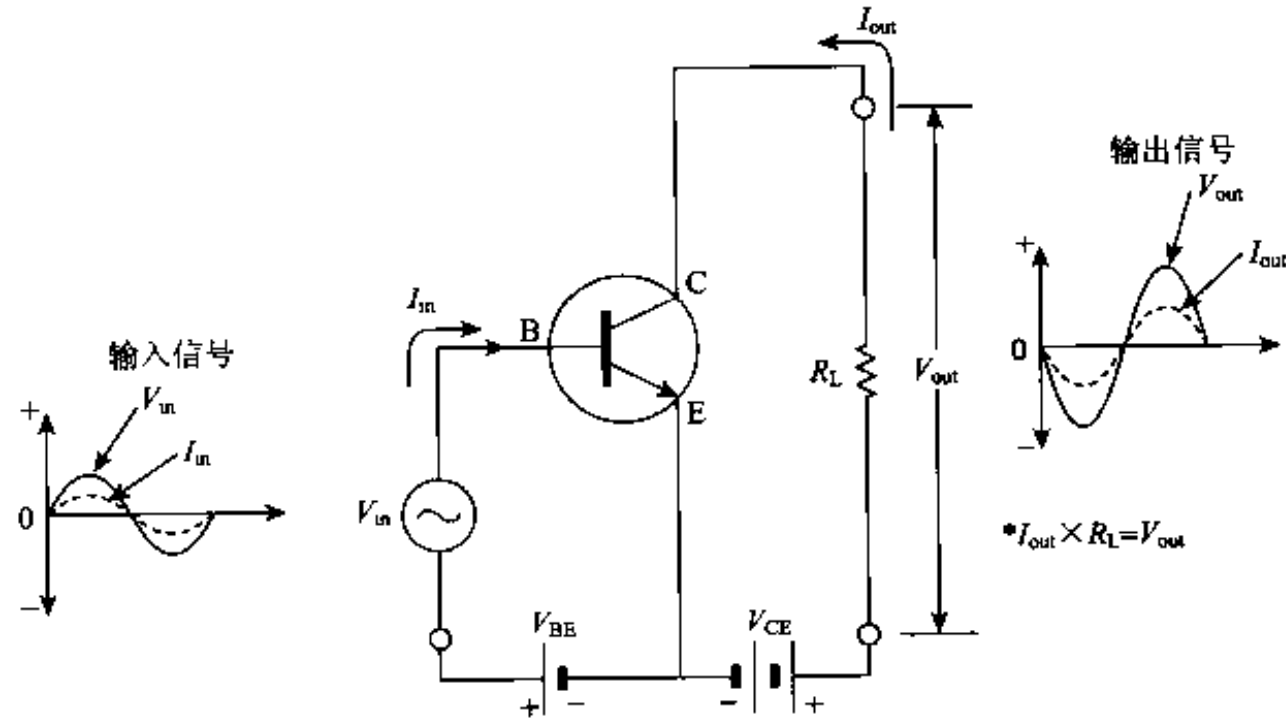
NPN型和PNP型之分,而在场效应型晶体管(FET)中则有n沟道型和p沟道型之分。

表 2.2 晶体管的种类和符号

种 类	符号图
双极型晶体管	NPN型晶体管
	PNP型晶体管
场效应型晶体管 (FET)	n沟道 MOSFET
	p沟道 MOSFET

如图 2.7 所示,双极型晶体管,用一个形象的比喻就好像从基极侧跑进 1 个人时,从集电极侧就能跑出 10 个人似的,也就是根据基极电流放大集电极电流。然而,能够跑进这里的人数,由其通道决定,所以是有界限的。即使基极电流流通,到一定限度后,集电极电流已不再增加了,这就是最大集电极电流。这样,在使用晶体管饱和区域工作时称为开关作用,可在电动机控制等领域中应用。

对于场效应晶体管,与栅极电压成正比的电流从漏极流向源极。与双极型晶体管与电流成正比相对应,FET 是与电压成正比。场效应晶体管由于具有低导通(ON)电阻及好的频率特性,所以最近在开关领域使用多起来了。



• 图 2.7 晶体管的基本电路和特性

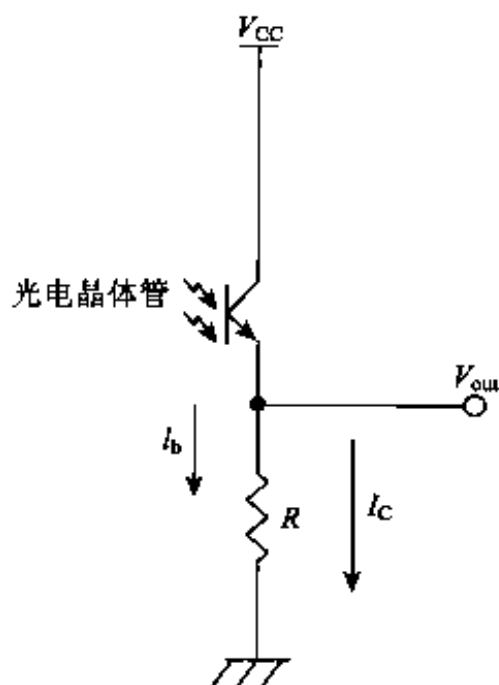
2.2.5 光电晶体管

光电晶体管的集电极电流不是根据基极电流而改变,它是根据光的强度改变集电极电流的。因此,可用在检测光强度的光传感器中。在机器人竞赛中,可以说所有机器人都要使用光传感器。光电晶体管的外形如照片 2.9 所示,管脚有 2 根。详细的情况在第 7 章中叙述。



• 照片 2.9 光电晶体管

按图 2.8 所示的电路图连接时,基极电流因光而流动。因此,可以获得与光强度成正比的输出电压 V_{out} 。



• 图 2.8 光电晶体管的基本电路

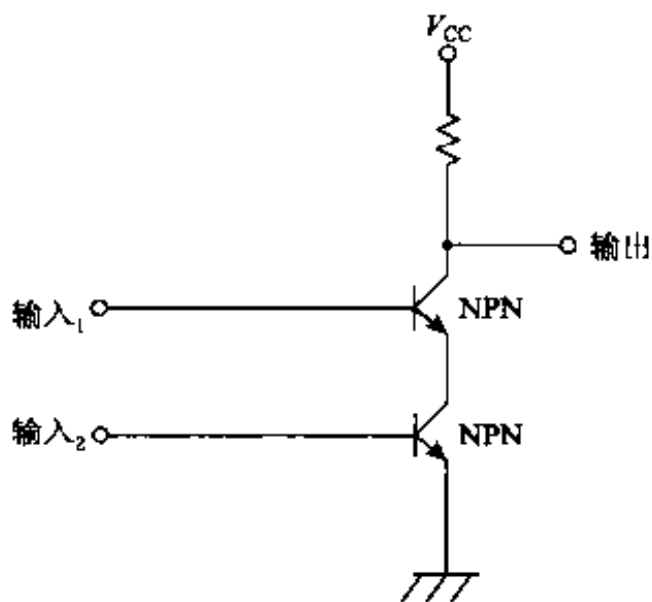
2.3 逻辑电路

逻辑电路有以晶体管-晶体管逻辑构成的 TTL 和以 CMOS 构成的两种。

逻辑运算在计算机领域称为布尔运算。布尔运算是由所谓的“0”和“1”两种符号决定的逻辑运算方法。这种运算包括 AND 和 OR,用这个电路实现的逻辑关系称为逻辑电路。将它们组装起来就构成计算机。

1) NAND 门电路

这是一种输入 1 和 2 同时为高电平(H)时,输出变为低电平(L)的电路。运算法则由于是 AND 和 NOT 的组合,所以称为 NAND(NOT + AND)电路(参见图 2.9)。

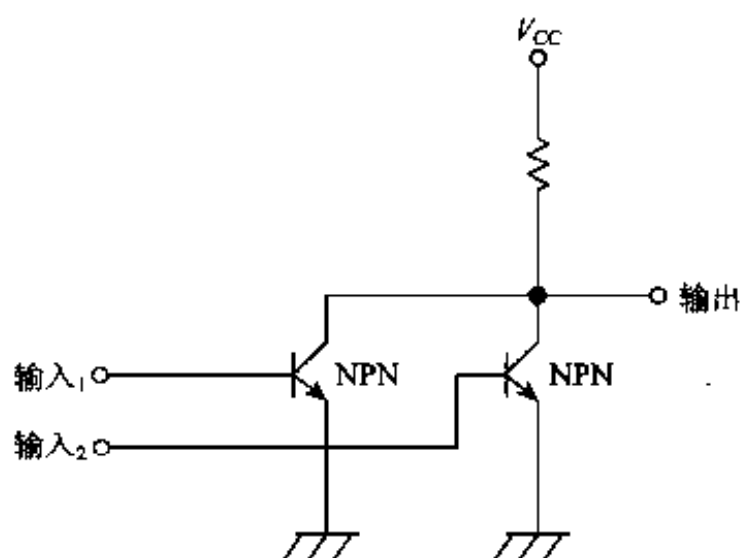


• 图 2.9 NAND 电路

2) NOR 门电路

是一种只要有一个输入为高电平,输出就变为低电平的电路,因为是

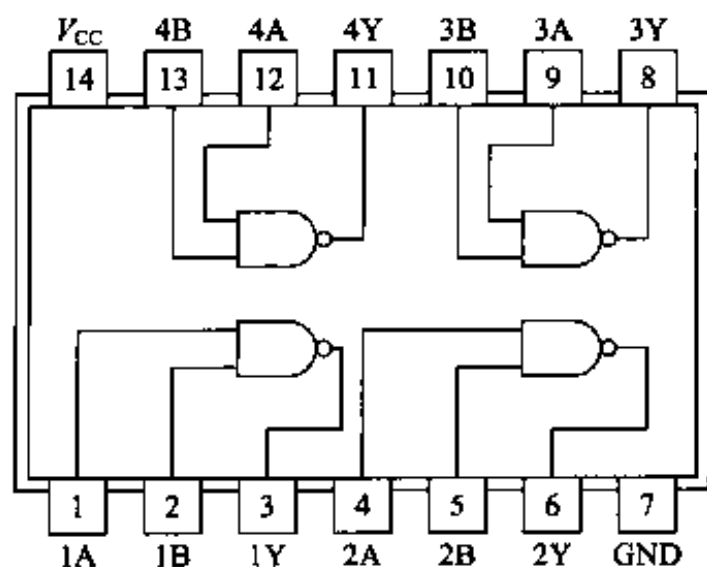
由 NOT + OR 组成的电路,所以称为 NOR 电路(参见图 2.10)。



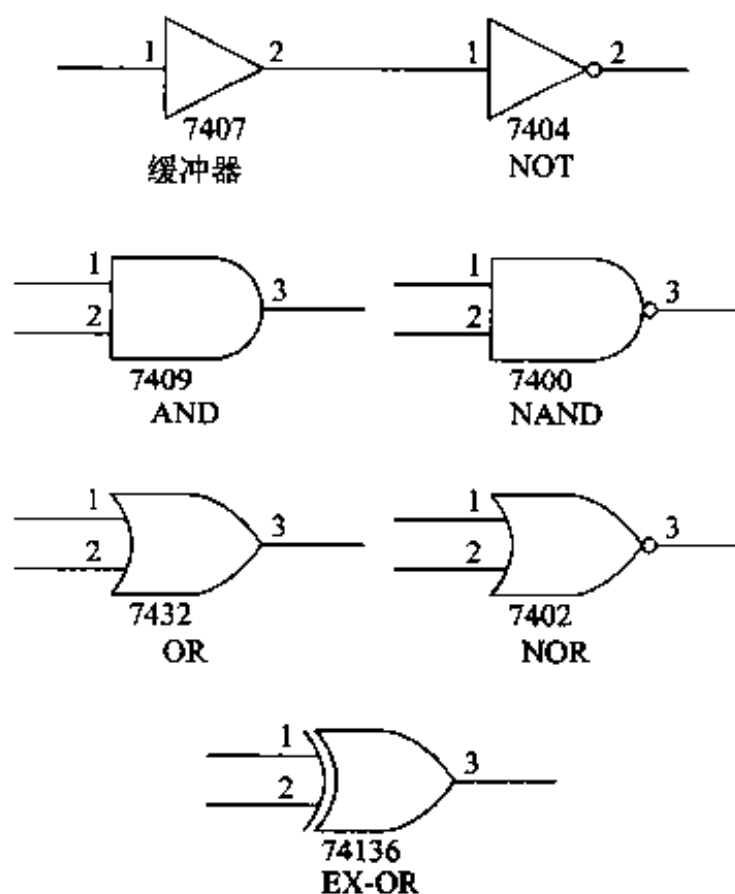
• 图 2.10 NOR 电路

2.3.1 TTL 集成电路

用分立晶体管构成上述电路很麻烦,若将几个这样的电路制作在芯片上封装在一起,就构成集成电路(IC),称为 TTL IC(参见图 2.11,2.12)。



• 图 2.11 TTL IC 实例(7400NAND)



• 图 2.12 其它 TTL 门电路(NOT, AND, OR, EX-OR)

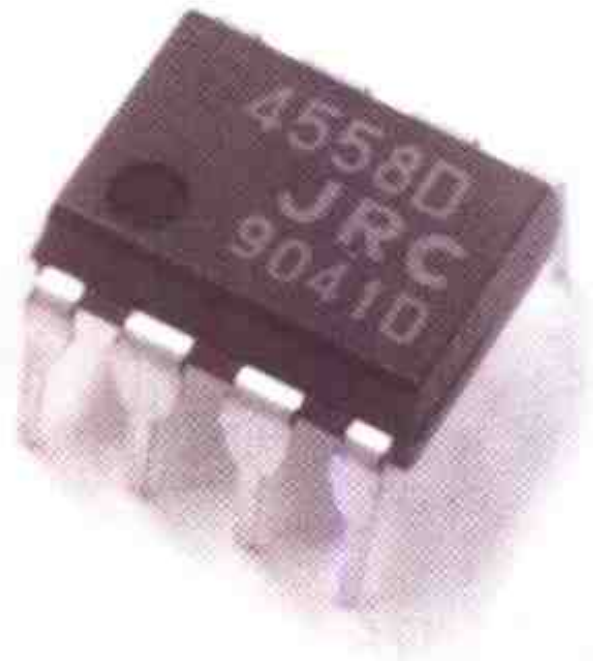
照片 2.10 所示的 4584 为 CMOS IC,在本书所介绍的机器人大赛中使用的就是这种集成块。



• 照片 2.10 CMOS IC(4584)

2.4 运算放大器

照片 2.11 所示的是运算放大器，可用于构成各种各样的放大器。下面给出其代表性实例。

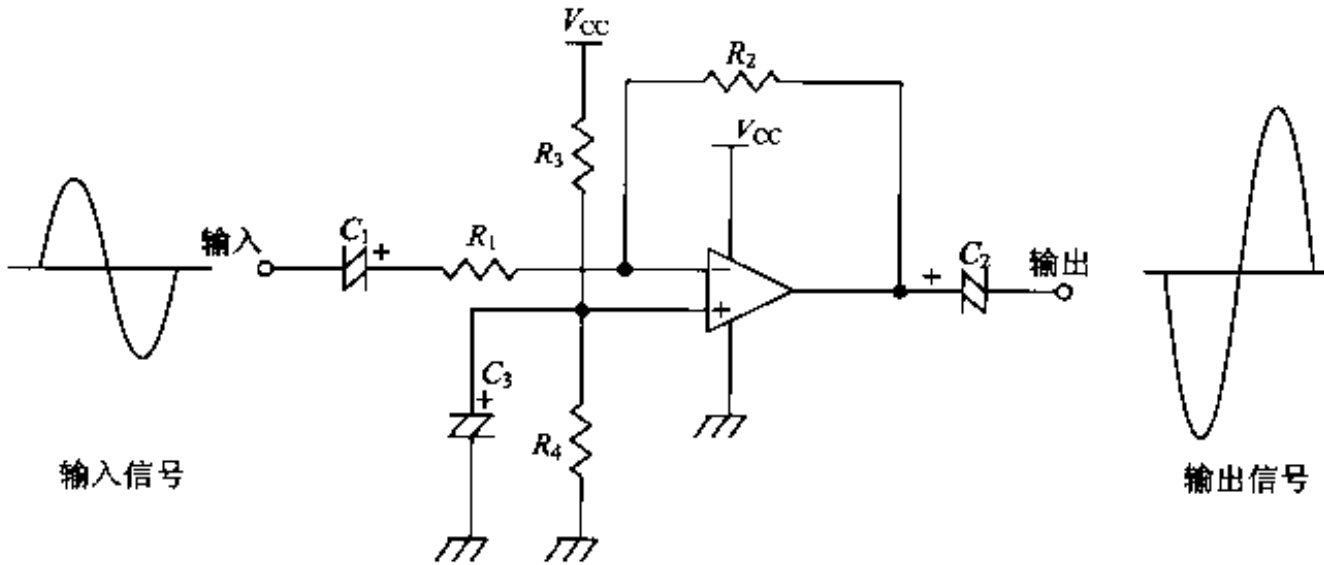


• 照片 2.11 运算放大器实例

对于机器人控制，将各种传感器的微小信号放大输入给计算机。例如输入给 H8CPU 的 A/D 转换器时，放大到满刻度 5V 左右，就能获得足够的分辨率。

在微型计算机中使用时，由于计算机电源多数情况用 5V 电压工作，所以可以使用 5V 单一电源，这时能使电路简化并且方便。因此，这里将介绍单一电源放大电路。

图 2.13 是用单一电源放大电路构成的 AC 放大电路，是反相放大器。



• 图 2.13 单一电源反相放大器

电源电压用电阻 R_3 、 R_4 分割，将中点电压设定在 V_{CC} 一半的地方，该电路对振幅对应于这一点上下摆动的输入

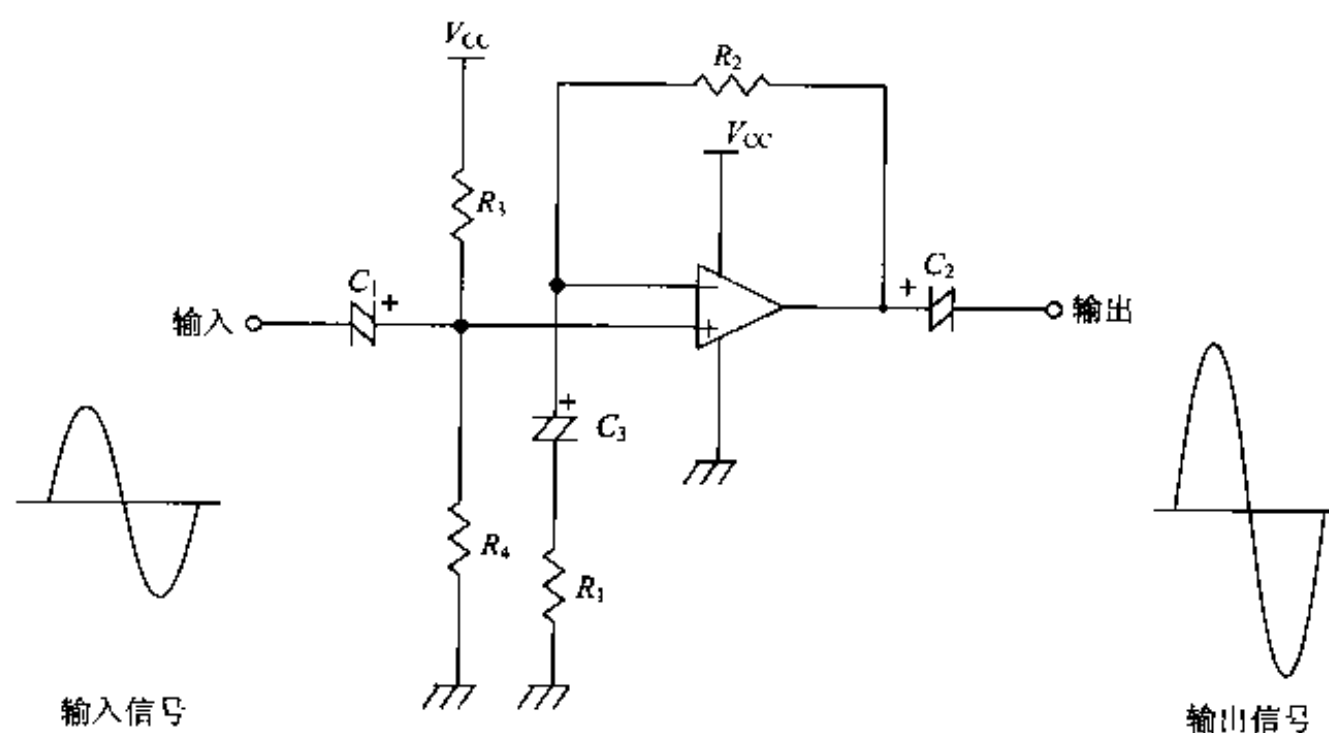
信号进行放大。这时的放大率变为 R_2/R_1 。由于输出对输入的相位反相，所以称为反相放大器。

低通截止频率为 $f = 1/2\pi C_1 R_1$ 。

用 A/D 转换器读入这个输出时，不要电容器 C_2 。微型计算机 A/D 的读

入一般就是在 0~5V 满刻度。

图 2.14 是正相放大器的实例，放大率变为 $(R_1 + R_2)/R_1$ 。



• 图 2.14 单一电源正相放大器

对反相放大器，各输入阻抗由输入侧的电阻 R_1 决定。由于这个值一变大放大率就不采用，所以阻抗完全由这个电阻决定。与其相对而言，对正相放大器，由于阻抗要与运算放大器相关，所以足够大就可以采用。因此，在传感器侧的内部电阻大的情况下，使用正相放大器时可得到小的损耗。

不只是运算放大器，一般的电路图，也有不画上电源的时候，所以即使电路图中未画电源也必需连接电源。另外，在电源上要加入旁路电容。

这些放大器可以在超声波传感器或光调制放大电路，以及回转仪传感器等的放大中使用。

有关运算放大器请参考以下文献：

《图解运算放大器电路》

高桥 宽主编 内山明治 村野 靖
著 OHM 社(中文版：陈镜超译，科学出版社，2000 年 1 月出版，2001 年 3 月第二次印刷)。

2.5 机器人设计工具简介

2.5.1 机械设计 CAD

有设计机器人用的 CAD 是很方便的。使用 CAD 研究机器人整体结构,然后以其作为基础,将各个零件的设计图纸画完,修改也简单。之后,打印输出,委托给合适的加工厂家即可。

在机械零件设计 CAD 中,有以下免费软件,可将它们下载使用。

1) MICROCADAM V1.3C

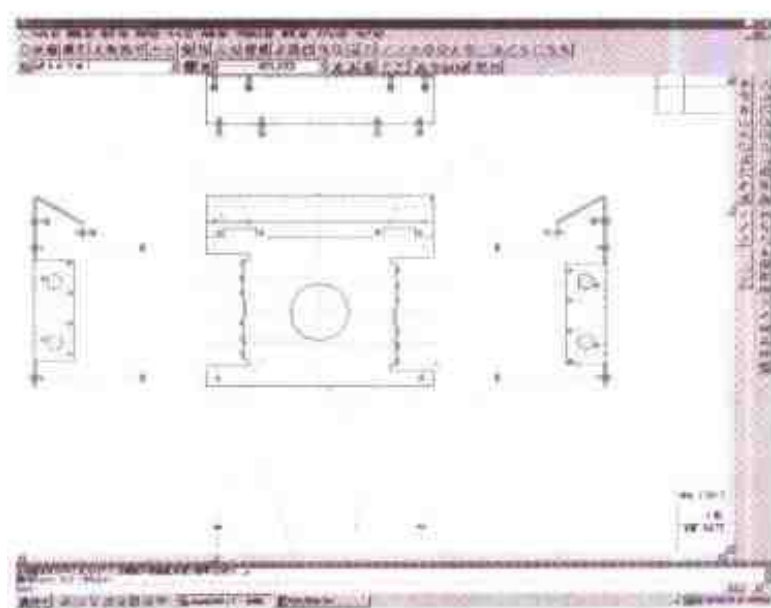
用 DOS 版软件,版本虽旧却是知名软件,又是不限定功能的实用版。

2) AutoCAD LT

由 Auto Desk 社提供了世界标准的设计软件作为二维 CAD 工具,即使对初学者也容易使用。AutoCAD LT

在日本也有出售,是价格低、使用方便的软件。实例版可以从以下的网址下载:

<http://www.autodesk.co.jp/prod-sup/acadlt/download/index.htm>



• 画面 2.1 AutoCAD LT 画面

2.6 电路设计、基板设计 CAD

对机器人,若有处理电路图或能设计基板的 CAD 是非常方便的。讨厌沾焊锡的人一定使用一下看。送给基板厂家 CAD 数据比送电路图还便于制作。

作为免费软件,介绍以下几家:

1) WinDraft 电路图编辑器

WinDraft 图解的拍摄(Schematic Capture)(仅 32 位)[3988kb](最新

版本 3.05 for Windows95/98/NT dated 23/3/2000)

<http://www.datamagic.co.jp/ivex/ivex.html>

用功能限定版,到 100 条引脚都可以使用。

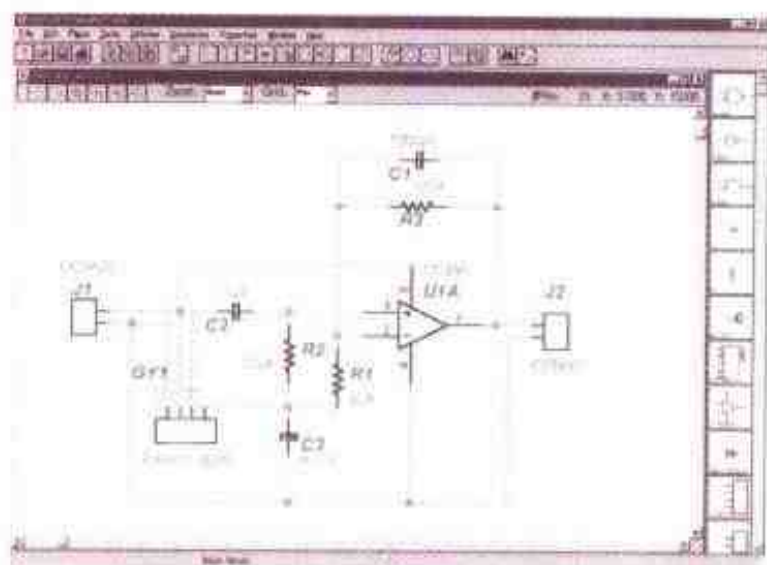
2) WinBoard PCB 线路布图

WinBoard PCB 线路布图 (Lay-

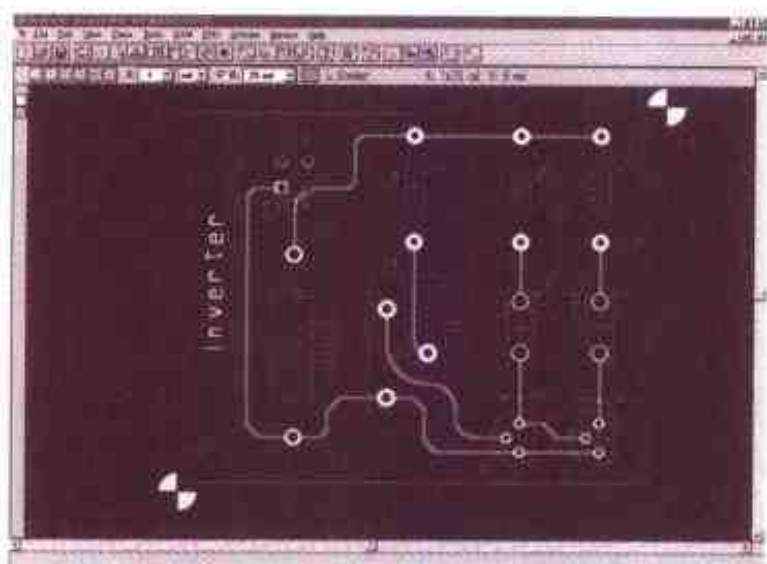
out)[3972kb](最后版本是 2.23 dated 17/3/2000)

<http://www.datamagic.co.jp/ivex/ivex.html>

用功能限定版,到 100 条引脚均可以使用。



• 画面 2.2 WinDraft 图解的拍摄

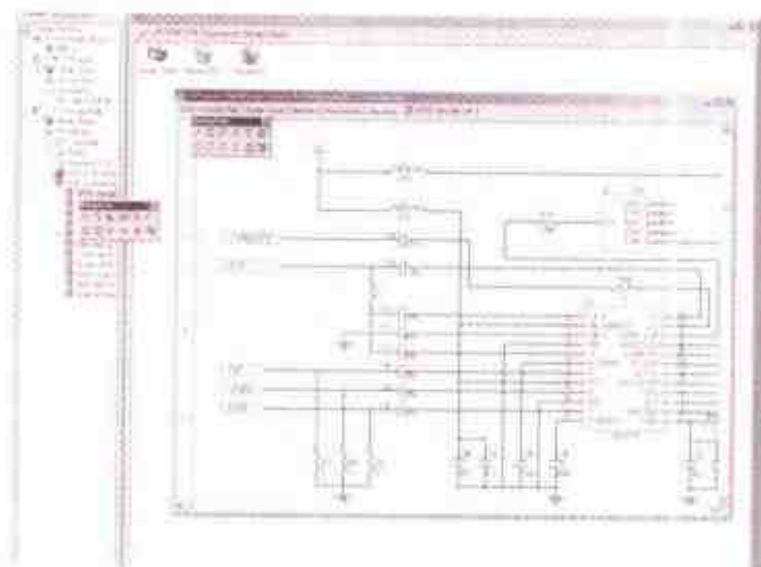


• 画面 2.3 WinBoard PCB 线路布图

3) Protel 99 SE Trial Version

Protel 电路设计和图形设计 CAD 试用版,可以从以下网址获得。或者送交 CD-ROM,限期为一个月。

<http://www.Protel.co.jp>



• 画面 2.4 Protel 99 SE 试用版画面

chapter

3

H8微型计算机

为了制作自律型机器人，必须有微型计算机。现在，微型计算机的应用，已经遍及到家电产品、移动电话、汽车等所有的领域。近年来，微型计算机已经不只是单纯进行计算处理的CPU（中央运算处理装置），通常都把为了运行程序所必需的存储器，与各种外部设备进行连接的输入输出端口（I/O）封装成一个插件。特别是日立制作所制作的H8系列微型计算机，根据每台微型计算机的使用目的（汽车用、家电用或工业用等），要准备许多不同功能的插件。

3.1 计算机

无论是从前还是现在,凡是称作计算机的,其基本的体系结构都一样。计算机就是只用 ON(1)和 OFF(0)两种状态的组合进行工作。ON 和 OFF 转换开关的种类,从继电器向半导体变化,从很大变为极小,从需要很大的空间,变为只有手指尖儿那么小。

习惯上以 ON 作为 1,以 OFF 作为 0 时,用一个开关表示状态为 0 或者为 1,只确定这两种状态。具有很多这种开关时,所表示的状态则为 2 乘以“开关数”,照这样乘下去,这种开关的个数就从 4 变成 8、从 8 变成 16、从 16 变成 32 及从 32 变成 64(位,二进制位),

随着时代的发展,这个二进制位变大起来。

· 2 进制和 16 进制数

我想大家能够理解,开关是计算机的基本功能,用 0 和 1 这两个数字表示,在世界上统称为 2 进制数,一般比我们使用的 10 进制数的位数多,并且,记述麻烦、转换难,所以通常使用称为 16 进制数的表示法。表 3.1 表示 10 进制数、2 进制数和 16 进制数的对应关系。从表中可以看出,16 进制数与 10 进制数的形式相似,但是,就本章所讲述的数值的意义几乎完全不能这样理解。

表 3.1 10 进制数、2 进制数和 16 进制数的对应表

10 进位法	2 进位法	16 进位法	汇编程序语言表示	C 语言表示
0	0	0	H'0	0×0
1	1	1	H'1	0×1
2	10	2	H'2	0×2
3	11	3	H'3	0×3
4	100	4	H'4	0×4
5	101	5	H'5	0×5
6	110	6	H'6	0×6
7	111	7	H'7	0×7
8	1000	8	H'8	0×8
9	1001	9	H'9	0×9
10	1010	A	H'A	0×a
11	1011	B	H'B	0×b
12	1100	C	H'C	0×c
13	1101	D	H'D	0×d
14	1110	E	H'E	0×e
15	1111	F	H'F	0×f

对 8 位(2 进制的 8 位)微型计算机,用 16 进制数 2 位表示,处理地址为 00~FF。对 16 位微型计算机,位数更进一步地增加。可以一次处理地址为 0000 ~FFFF。而且,在下一章以后

出现的汇编程序语言或 C 程序语言那样的程序设计语言中,表示 16 进制数时,对汇编程序,在数值终结的前部附加如 FFH 或 H'FF 符号,对 C 语言程序用 0xff 记述。

3.2 H8-3048F 单片机

如前所述,最近的微型计算机,不仅是 CPU,而且增加 ROM 或 RAM、I/O 等装配在一个芯片上。单纯用眼睛看,不可能说哪一部分是 CPU、哪一部分是 I/O。照片 3.1 是本书中使用的日立微型计算机 H8-3048F 的芯片。它基本上是一个四方形的,并且在周围出来无数个像蜈蚣腿那样引脚的大规模集成电路(LSI)。即使如此,10 年前如果制作出同样功能的微型计算机,至少需要十倍的空间。



• 照片 3.1 单片机

日立 H8 CPU 系列在机器人领域已成为经常使用的微型计算机。特别

是 H8-3048F 是日本微型计算机圣杯机器人竞赛采用的微型计算机。大会会场要布置很多微型计算机插件板。当然,在其它的机器人竞赛中也多采用 H8 单片机。

图 3.1 表示这种微型计算机内部的结构图。I/O 端口、A/D 转换器、D/A 转换器和计时器等各种各样功能装入一个芯片中。全部说明这些功能,就能写出一本书了,所以这里仅仅将大体结构部分进行简单的介绍。

• CPU

CPU 称为中央运算处理装置,它根据指令忠实地执行处理,可以说相当于人类的大脑。

• ROM

ROM(只读存储器)为通常仅能从 CPU 读出的存储器(记忆装置)。用户事先在 ROM 上写好程序,CPU 将指令从 ROM 中读出使用。H8-3048F 内装快速存储器(Flash ROM),利用 CPU 的功能,采用从个人电脑上用通信改写 ROM 数据的方式。即使计算机的电源断电,存储下来的内容也不会消失。

• RAM

RAM(随机存取存储器)为可以一边任意地写入、一边任意地删除的存储器。CPU 在某种场合可以使用存储的数据。微型计算机断电的同时存储下来的内容完全消失。

• 输入/输出端口

输入/输出(I/O)端口是微型计算机对外能输入数据或输出数据的端口。可以进行连接端口的开关状态的读入或进行显示器(LED)的开关。

• A/D 转换器

A/D 转换器是将模拟量转换为数字量的装置。对 CPU, 为了作为数值使用所有的数据, 和人们读取测试器的刻度一样, 将 A/D 转换引脚的输入电压数字化。H8-3048F 单片机可以使用 8 通道的 A/D 转换器。

• D/A 转换器

D/A 转换器是用计算机将处理的数字量转换成模拟量的装置。和 A/D 转换器相反, 由 CPU 将处理的数值从 D/A 转换器的引脚作为电压输出。H8-3048F 单片机可以使用 2 通道的 D/A 转换器。

• 脉冲输入

脉冲输入可以测量 ON 和 OFF 的重复次数。本书中连接旋转编码器, 计算旋转编码器旋转时产生的 ON/OFF 脉冲序列, 利用它测量距离或测量速度等。

• PWM 输出

PWM 就是脉宽调制, 可以使用它控制电动机的速度等。对图 3.1 所示 CPU, 最大可达到 5 通道的 PWM 输出。

• 串行通信端口

是保持串行通信的部分。对图 3.1 所示 CPU 能达到 2 通道。H8 的内置快速存储器(Flash ROM)写入时也使用这个端口。

下面, 介绍硬件操作的要点。详细情况请参照 H8-3048F 单片机硬件操作说明书, 这里仅就在机器人中使用的寄存器进行简单地讲解。

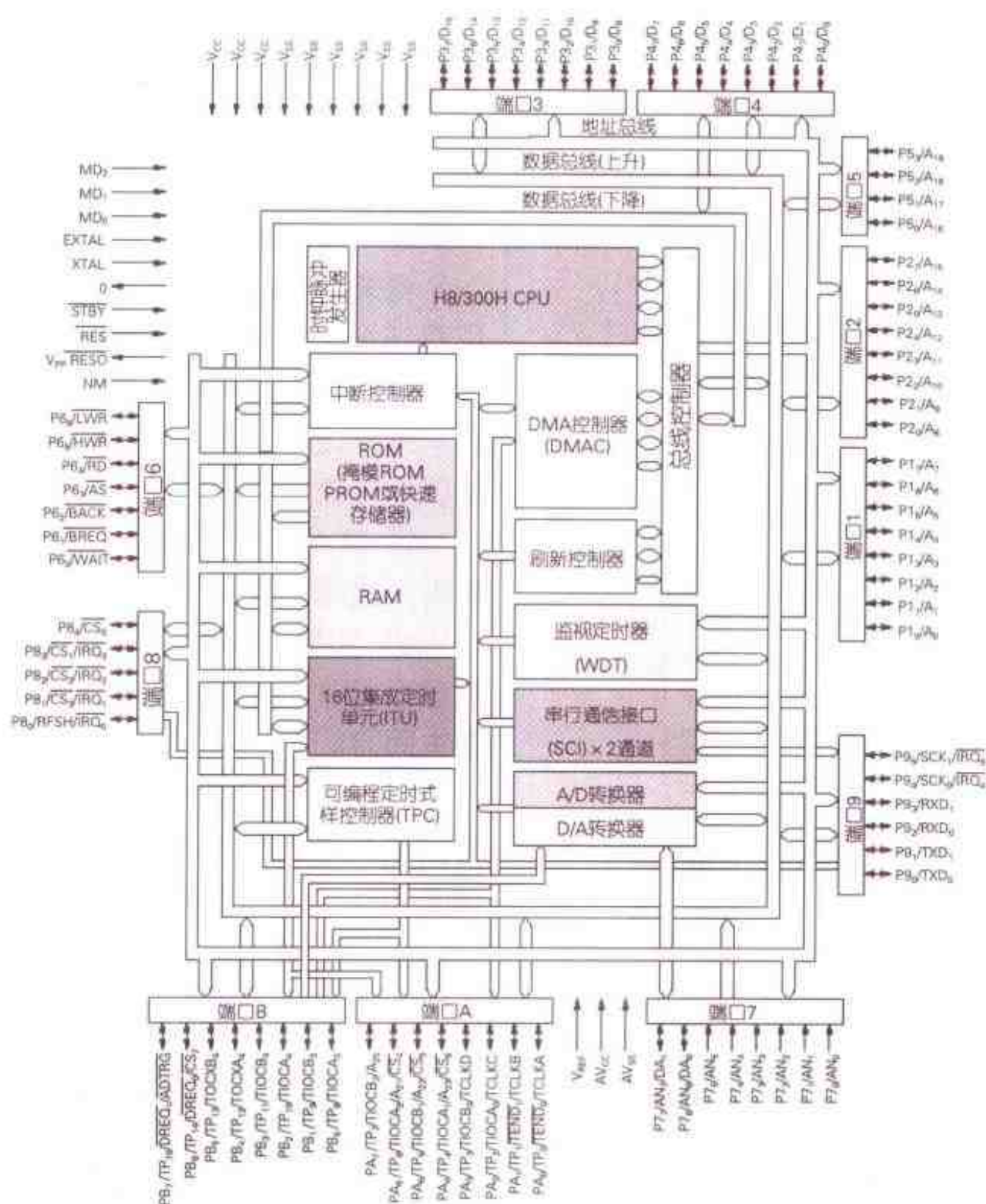
3.2.1 CPU 运作模式的设定

图 3.1 所示 CPU 中有 7 种运作模式(模式 1~7)。这些模式可以通过模式引脚(MD₂ ~ MD₀)设定。根据这个设定, 可以选择总线方式的初始方式和地址空间。地址空间可以选择 1M 字节, 也可以选择 16M 字节。

为了在模式 7 中使用, 将 MD₂ ~ MD₀ 通过 10kΩ 左右的电阻连接 5V 电压。模式 7 称为单片机先进(advance)模式, 内装 ROM 128k 字节及内装 RAM 4k 字节, 总线宽度为 16 位, 对于这种方式, 使用专用的端口。

图 3.2 表示将 H8-3048F 设定模式 7 时的存储器地址映像。图中左侧写的是存储器的地址。在 H'00000 ~ H'1FFFF 中内置快速存储器(Flash ROM), 在 H'FEF10 ~ H'FFF0F 中内置 RAM。这种分配完全是在 H8-3048F 设计阶段决定的, 以后不能改变。

在内置 ROM 区域的最前面是向量区域, 当发生中断时, 它记述所有跳转地址。有关中断在以后说明, 这里只希望大家记住有“中断”这个词。



• 图 3.1 H8-3048F 单片机内部结构图

· 输入/输出端 ☐

图 3.1 所示框图中,标注“端口”部分的都是输入/输出端口。例如,相扑机器人情况,可以使用摔跤场地或对手传感器作为输入,控制电动机信号作为输出。

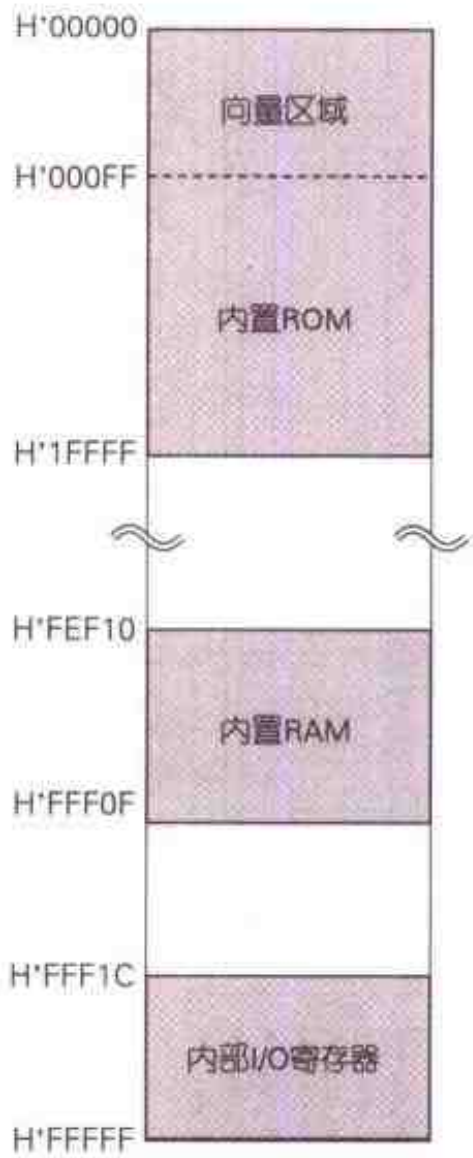
为处理这些端口的各种设定以及由 CPU 处理输入/输出数据的寄存器

就是图 3.2 中的内部 I/O 寄存器。和 RAM 完全相同,不妨看作是从 CPU 读写的一种存储器。

将设定输入或者输出端口的输入/输出方向的寄存器称作数据定向寄存器(DDR),并且实际上,将又向端口输出信号,又读出信号的寄存器称为数据寄存器(DR)。

表 3.2 端口 2、6 的 DDR、DR 的寄存器地址

端口名	DDR 地址	DR 地址
端口 2	H'FFFC1	H'FFFC3
端口 6	H'FFFC9	H'FFFOB



• 图 3.2 存储器地址映像

例如,把端口 2 作为输出、端口 6 作为输入时,各相应的寄存器地址如表 3.2 所示,初始设定,通过在端口 2

的 DDR 上写入 H'FF (在 H'FFFC1 地址上写入 H'FF),可以指定输出端口。另外,在端口 6 的 DDR 上如果写入 H'00,可以作为输入端口使用,数据的写入或读入通过各个 DR 寄存器进行。

有关这些寄存器的详细功能及设定方法在 H8-3048F 单片机硬件说明书中有详细记述,所以请一定参考去做。另外,在阅读操作说明书时,希望从以下几点入手:

- ① 操作说明书既厚又难懂,希望技术人员能化繁为简地考虑;
- ② 我想不必全部理解微型计算机理论,仅以制作机器人必需的功能为主即可。

这里,以能使机器人动作为重点,所以最初不要弄得太复杂,有充裕时间时,可仔细地阅读硬件操作说明书(参照照片 3.2)。



• 照片 3.2 H8-3048F 硬件操作说明书

3.3 集成定时单元 (ITU)

图 3.1 所示 H8-3048F 内部结构图中的 ITU 称为集成定时单元,限于篇幅这里不能详细讲解,仅就与机器人有关的 3 部分功能——PWM 模式、

计相模式及输入捕捉进行说明。

3.3.1 PWM

控制电动机转速时,通常是用 PWM 控制。在使用 H8 的 PWM 输出

功能时,使用表 3.3 所示寄存器。ITU 有 5 通道,这里仅采用 0 通道,而剩余的 1~4 通道仅仅是地址不同,其它的通道也有相同寄存器。

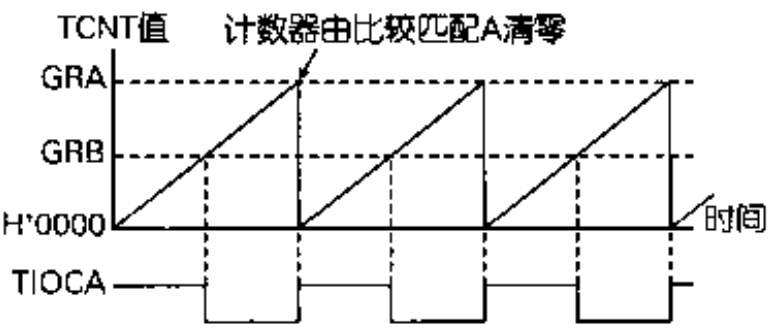
表 3.3 与 ITU 有关的寄存器

通 道	地 址	名 称	简 称
共 同	H'FF60	定时器启动寄存器	TSTR
	H'FF61	定时器同步寄存器	TSNC
	H'FF62	定时器方式寄存器	TMDR
	H'FF63	定时器功能控制寄存器	TFCR
	H'FF90	定时器主输出允许寄存器	TOER
	H'FF91	定时输出控制寄存器	TOCR
0	H'FF64	定时器控制寄存器 0	TCR0
	H'FF65	定时器 I/O 控制寄存器 0	TIOR0
	H'FF66	定时器中断允许寄存器 0	TIER0
	H'FF67	定时状态寄存器 0	TSR0
	H'FF68	定时计数器 0H	TCNT0H
	H'FF69	定时计数器 0L	TCNT0L
	H'FF6A	通用寄存器 A0H	GRA0H
	H'FF6B	通用寄存器 A0L	GRA0L
	H'FF6C	通用寄存器 B0H	GRB0H
	H'FF6D	通用寄存器 B0L	GRB0L

1. PWM 模式工作原理

图 3.3 中示出 H8 PWM 模式的工作原理。设定 PWM 模式时,则 TIOCA 引脚为输出,当 GRA 比较匹配时,TIOCA 输出 1、当 GRB 比较匹配时,TIOCA 输出 0。上述在硬件操作说明书中均有说明。讲解它时,有所谓 TCNT 计数器(在某一定时间间隔能自动计数的存储器),如果预先在 GRB 寄存器写入值和 TCNT 的数字相同,则 TIOCA 的输出端为 0。并且,增加 TCNT 计数值时,在另外某一个 GRA 寄存器预先写入的值和其相同时,使

TIOCA 端为 1,同时也使 TCNT 的值为 0。换句话说,如果事先使用所谓 GRA 和 GRB 的寄存器,设定将 TIOCA 端变成 1 和变成 0 的时刻,就能输出 PWM 信号。



• 图 3.3 PWM 的工作原理

2. PWM 模式的设定

PWM 模式的设定步骤如下:

① 通过 TCR 寄存器中的 TPSC2 ~ TPSC0 位(二进制位)选择计数器时钟脉冲;

② 通过 TCR 寄存器的 CCLR1、CCLR0 位(二进制位)选择计数器清 0 源;

③ 在 GRA 寄存器中设置 PWM 波形变为 1 的输出时间值;

④ 在 GRB 寄存器中设置 PWM 波形变为 0 的输出时间值;

⑤ 设 TMDR 寄存器 PWM 位(二进制位)置 1, 设定 PWM 方式;

⑥ 开始启动定时器计数工作。

3.3.2 各种寄存器

仅介绍设定 PWM 方式需要的寄存器。

1. 定时模式寄存器(TMDR)

在定时器中使用 PWM 模式时,应根据把相当于定时器模式寄存器(TM-DR)中使用的定时器的 0~4 位及 6 位(二进制位)置为 1, 来设定 PWM 方式(参见图 3.4)。

并且,仅在通道 2 有称为相位系数模式(设 TMDR 寄存器位 6 为 1)的特殊模式,自动记下 2 相旋转编码器的脉冲,将这个值写入 TCNT 寄存器。对于 2 相的场合根据编码器的旋转方向,计数上升或下降,因此可以在决定相对位置等的控制中使用。

2. 定时控制寄存器(TCR)

定时控制寄存器(TCR)选择可还

原计数的时钟(参见图 3.5)。

用位(二进制位)2 选择使用外部时钟还是使用内部时钟,这里使用内部时钟所以选择为 0(参见表 3.4)。

表 3.4 计数器预标度(prescaler)

位 1	位 0	说 明
0	0	用内部时钟 ϕ 计数
0	1	用内部时钟 $\phi/2$ 计数
1	0	用内部时钟 $\phi/4$ 计数
1	1	用内部时钟 $\phi/8$ 计数

H8-3048F 的最大输入时钟为 16MHz, 所以连接 16MHz 晶体信号传输器时的分辨率分别如表 3.5 所示。

表 3.5 1 个计数的时间

位 1	位 0	1 个计数的时间
0	0	$1/16\mu\text{s}$
0	1	$1/8\mu\text{s}$
1	0	$1/4\mu\text{s}$
1	1	$1/2\mu\text{s}$

另外, TCNC 寄存器为 16 位(用 10 进制数处理 0~65535 范围的数值)计数器, 所以如果分别乘以 216 就可以获得满计数时间。如表 3.6 所示。

表 3.6 满计数时间

位 1	位 0	满 65535 计数的时间
0	0	4.096ms
0	1	8.192ms
1	0	16.384ms
1	1	32.768ms

因而, 将位(二进制位)1 和 0 分别设定为 1, 如果是 20000 个计数, 刚好

可以测量 10ms。

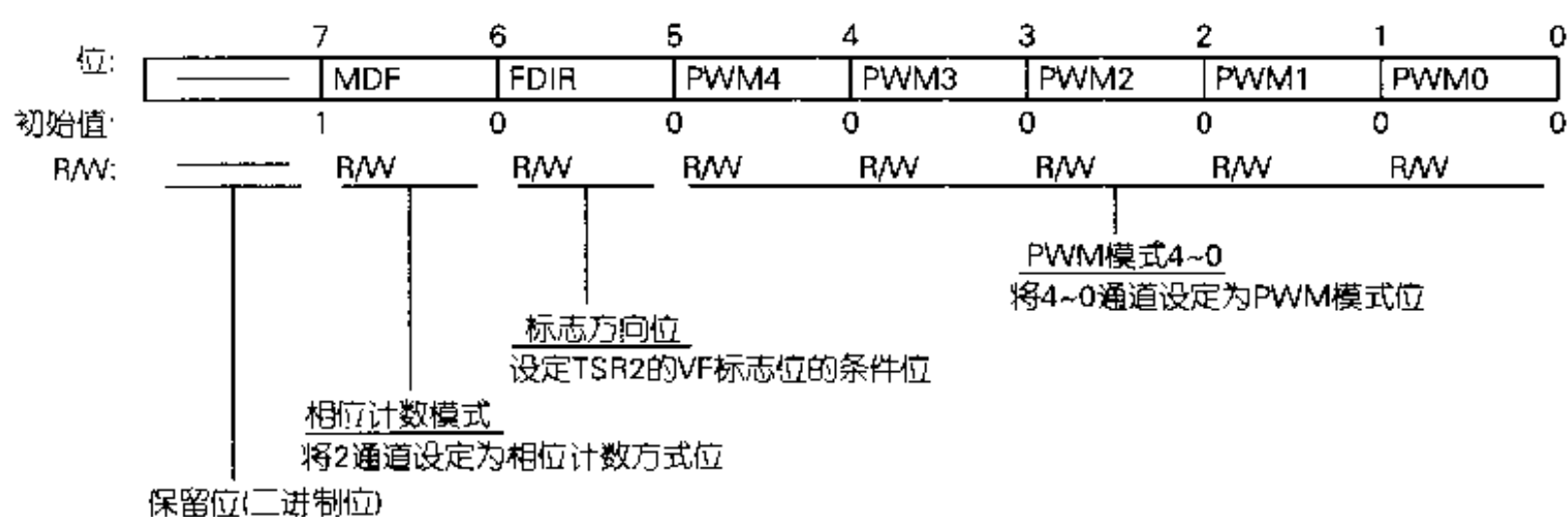
位(二进制位)3、4 时钟脉冲波形决定于在什么地方进行计数。0 为脉冲上升沿、1 为脉冲下降沿,定时检测它们递增计数。这里,选择脉冲上升沿,分别设定为 0。

位(二进制位)5、6 取决于清 0 计数器的因子。其中,选择和 GRA 比较

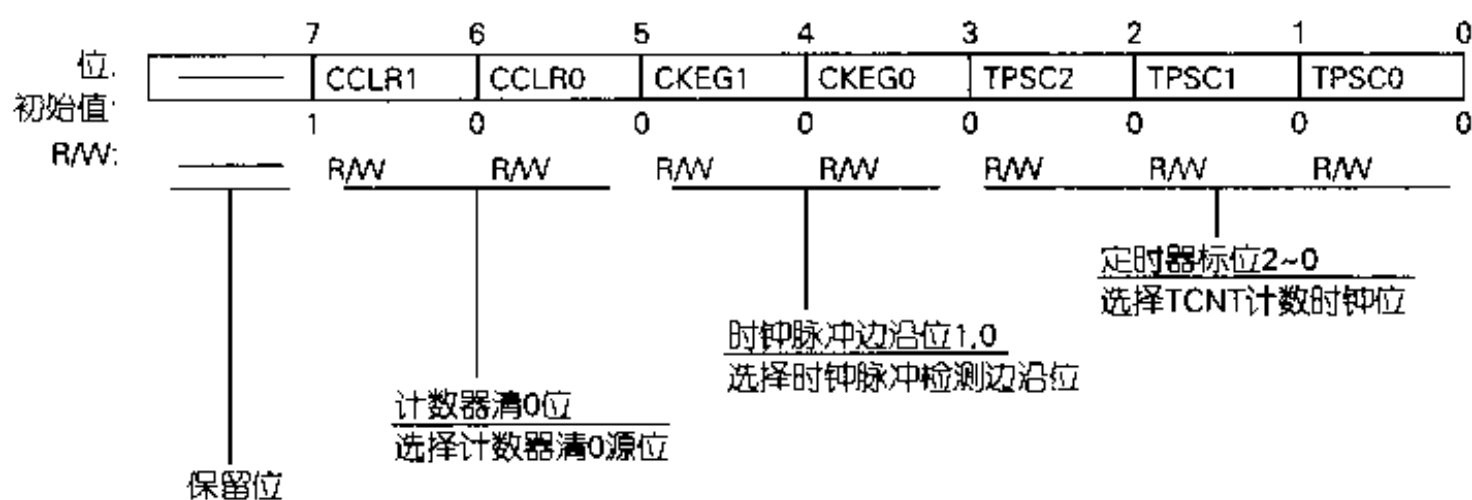
匹配的清 0 因子,分别设定为 0 和 1。

3. 定时启动寄存器(TSTR)

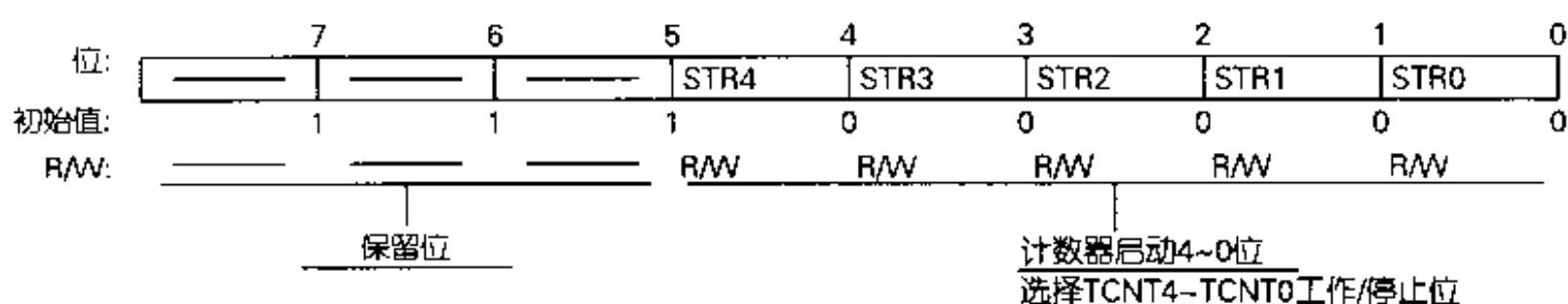
定时启动寄存器(TSTR)负责 0~4 通道的 TCNC 工作或者停止。如果将各个位(二进制位)置 0,则各定时器停止工作,如果设为 1 则开始计数(参见图 3.6)。



• 图 3.4 定时器方式寄存器(TMDR)



• 图 3.5 定时控制寄存器(TCR)



• 图 3.6 定时器启动寄存器(TSTR)

3.3.3 相位计数模式

下面讨论为了将从旋转编码器输出的信号写入定时器的各种模式。

如前所述,相位计数模式检测 2 个外部时钟输入(TCLKA、TCLKB)引脚的相位差,TCNT2 根据检测结果相应进行递增或递减计数。

1. 相位计数模式设定步骤实例

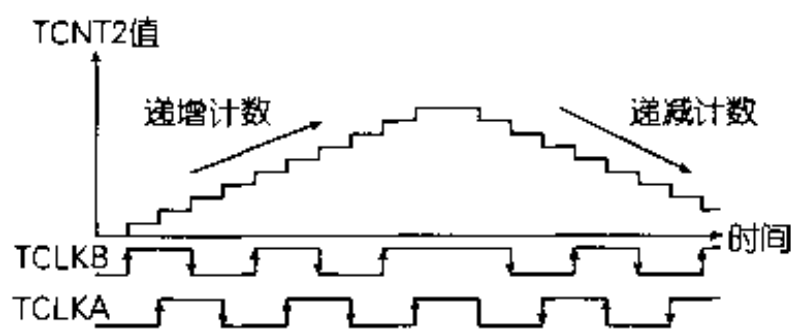
① 将 TMDR 的 MDF 位(二进制位)置 1,选择相位计数模式。

② 用 TMDR 的 FDIR 位(二进制位)选择标志置位条件。

③ TSTR 的 STR2(二进制位)置 1,计数器开始工作。

2. 相位计数模式的工作原理

图 3.7 中示出了相位计数模式的工作实例,表 3.7 中分别表示 TCNT2 的递增/递减计数条件,对相位计数模式,在 TCLKA、TCLKB 引脚的脉冲上升/下降沿两个边沿都要计数。因而,连接每转输出 100 个脉冲的旋转编码器时,一转产生 200 个脉冲。



• 图 3.7 相位计数模式设定

表 3.7 递增、递减计数条件

计数方向	递增计数				递减计数			
TCLKB	≠	高	≠	低	高	≠	低	≠
TCLKA	低	≠	高	≠	高	低	≠	高

3.3.4 输入捕捉

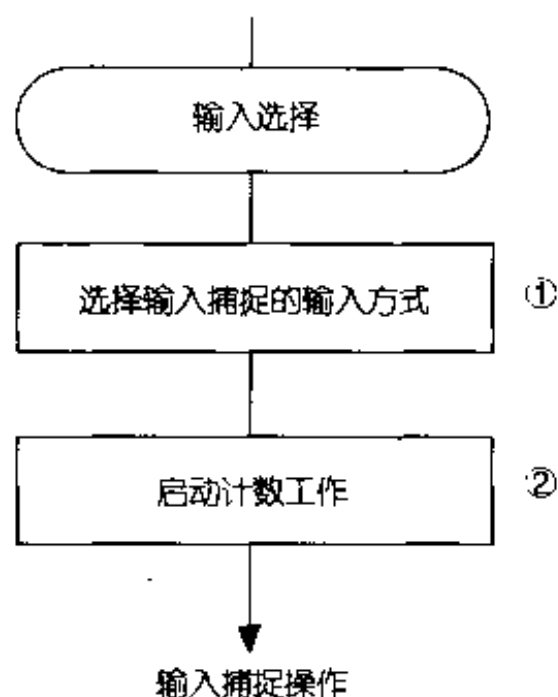
1. 输入捕捉功能

利用输入捕捉功能,可以进行脉冲宽度或周期的测定。

检测输入捕捉/输出比较引脚(TIOCA、TIOCB)的输入脉冲信号上沿或下沿,可以将 TCNC 的值传送给 GR。检测沿可以从上升沿/下降沿两沿中选择。

2. 输入捕捉操作的设定步骤

输入捕捉操作的设定,在 TMDR 及 TCR 设定后,按照图 3.8 的步骤进行。



• 图 3.8 输入捕捉操作的设定步骤

首先,通过 TIOR 将 GR 设定在输入捕捉寄存器上,将输入捕捉信号的输入沿从其上升沿/下降沿及两沿 3 种类型中选择。其中,将对应点的 DDR 以清 0 方式进行 TIOR 的设定。

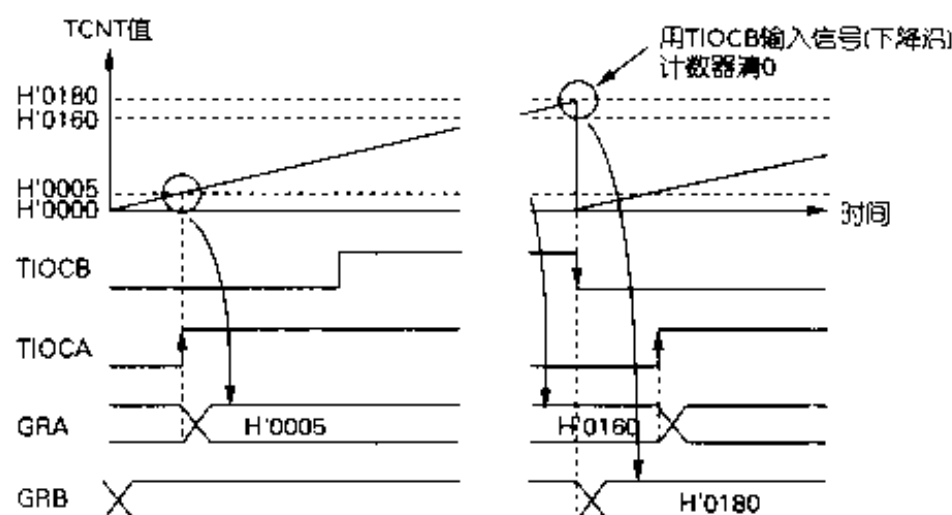
其次,将 TSTR 的 STR 位(二进制

位)置 1, TCNT 启动计数。

3. 输入捕捉动作实例

图 3.9 中示出输入捕捉动作实例。TIOCA 引脚的输入捕捉输入信号

沿选择上升/下降两沿, 另外, TIOCB 引脚的输入捕捉输入沿选择下降沿, TCNT 为 GRB 输入捕捉设定计数器清 0 时的实例。



• 图 3.9 输入捕捉操作实例

3.4 A/D 转换器、D/A 转换器

3.4.1 A/D 转换器

A/D 转换器(ADC)有 8 个模拟通道, 分 0、1 两组, 每组各 4 个通道可以使用(参见表 3.8)。

表 3.8 各通道和 A/D 数据寄存器

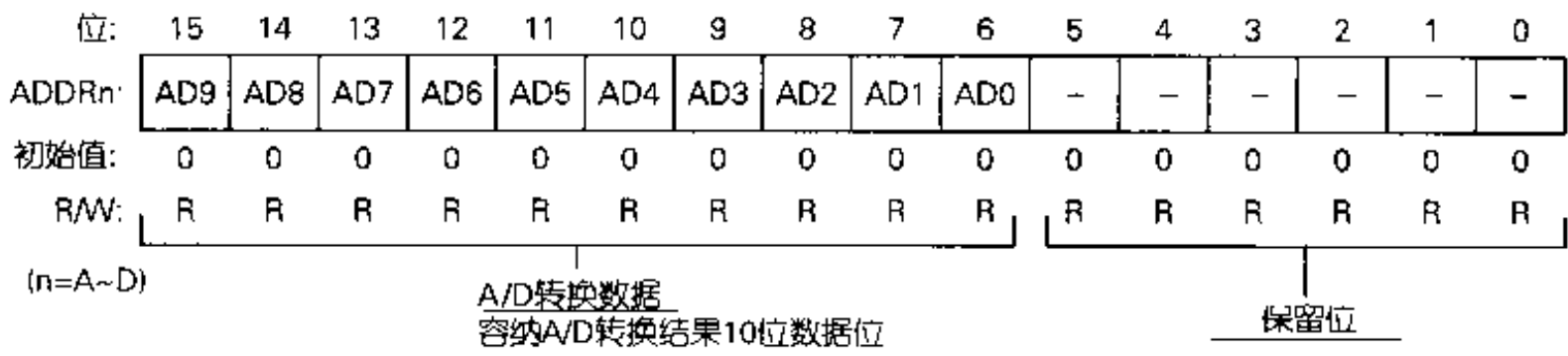
模拟输入通道		A/D 数据寄存器
0 组	1 组	
AN ₀	AN ₄	ADDRA
AN ₁	AN ₅	ADDRB
AN ₂	AN ₆	ADDRC
AN ₃	AN ₇	ADDRD

A/D 转换器的设定通过图 3.11

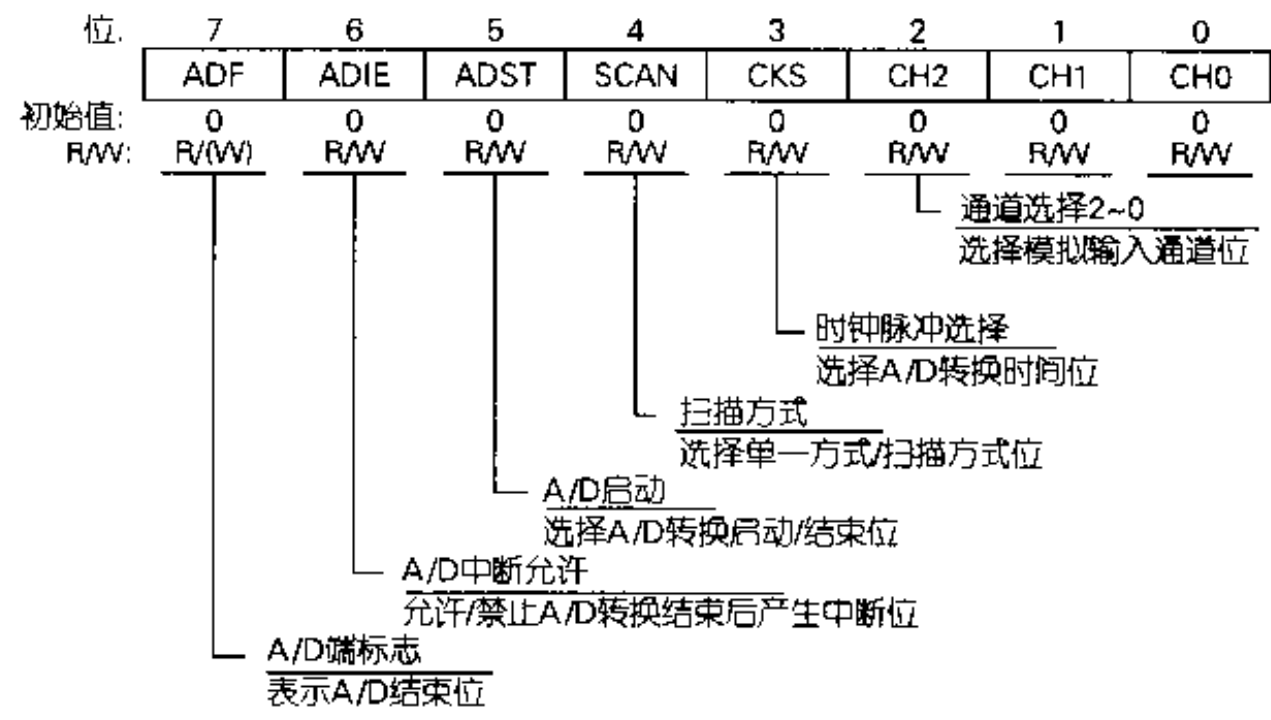
中的 A/D 控制状态寄存器(ADCSR)进行。

首先, 根据表 3.9 设定通道转换。单一模式为仅 1 个通道进行的 A/D 转换方式, 对扫描模式为几个通道连续进行的 A/D 转换方式。这里用 ADCSR 的位 4(二进制位)选择。以 0 为单一模式, 1 为扫描方式。

例如, 如果设定 0~3 通道扫描通道, 仅仅将 ADCSR 的 A/D 起始位(位 5)置 1, 如果希望读出数据寄存器 ADDRA、ADDRB、ADDRC 和 ADDR D 的数据, 输入 AD 转换完的新数据即可。



• 图 3.10 A/D 数据寄存器(ADDR0~7)



• 图 3.11 A/D 控制状态寄存器(ADCSR)

表 3.9 ADCSR 的通道选择(位 2~0)

选择组	选择通道		说 明	
CH2	CH1	CH0	单一方式	扫描方式
0	0	0	AN ₀ (初始值)	AN ₀
	0	1	AN ₁	AN ₀ 、AN ₁
	1	0	AN ₂	AN ₀ ~ AN ₂
	1	1	AN ₃	AN ₀ ~ AN ₃
1	0	0	AN ₄	AN ₄
	0	1	AN ₅	AN ₄ 、AN ₅
	1	0	AN ₆	AN ₄ ~ AN ₆
	1	1	AN ₇	AN ₄ ~ AN ₇

3.4.2 D/A 转换器

将数字量转换成模拟量的装置为D/A转换器。转换器引脚如表3.10所示,可以使用DA₀、DA₁ 2通道。但是这个接线引脚和A/D转换器共用,将D/A通道两方面使用时,A/D转换器不能使用2通道部分。

通过设定图3.12的DACR及图

3.13的DASTCR寄存器,可以使用D/A转换器。

用DA₀E₁、DA₀E₀控制各通道输出、转换位,如表3.12所示,它和DAE的组合可以控制D/A转换。

使DASTE位(0位)置1,则用设定DADR寄存器的值可以从D/A引脚输出电压。

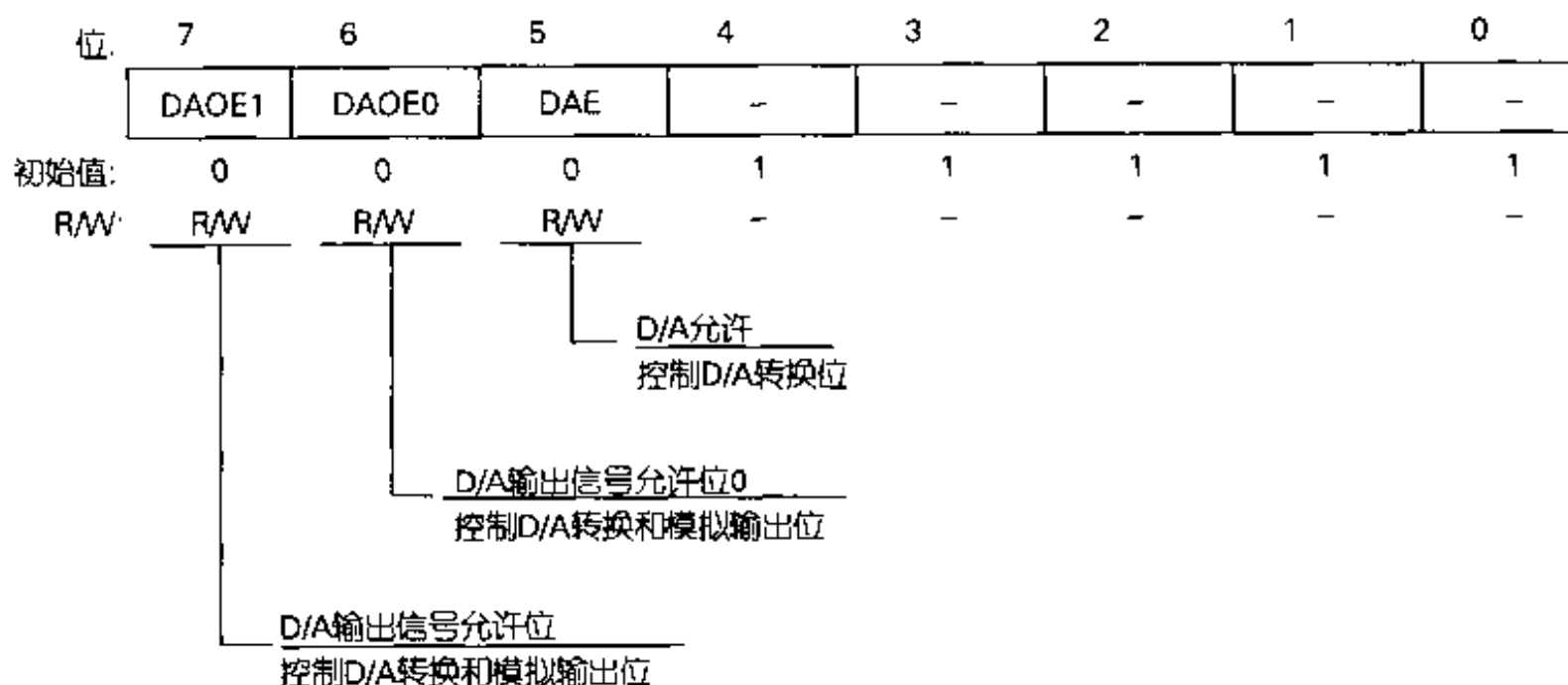
表 3.10 D/A 转换器引脚的构成

引脚名	简称	输入/输出	功 能
模拟电源引脚	AV _{CC}	输入	模拟部分的电源及基准电压
模拟接地引脚	AV _{SS}	输入	模拟部分的接地及基准电压
模拟输出引脚 0	DA ₀	输出	通道 0 模拟输出
模拟输出引脚 1	DA ₁	输出	通道 1 模拟输出
基准电压引脚	V _{REF}	输入	模拟部分基准电压输入

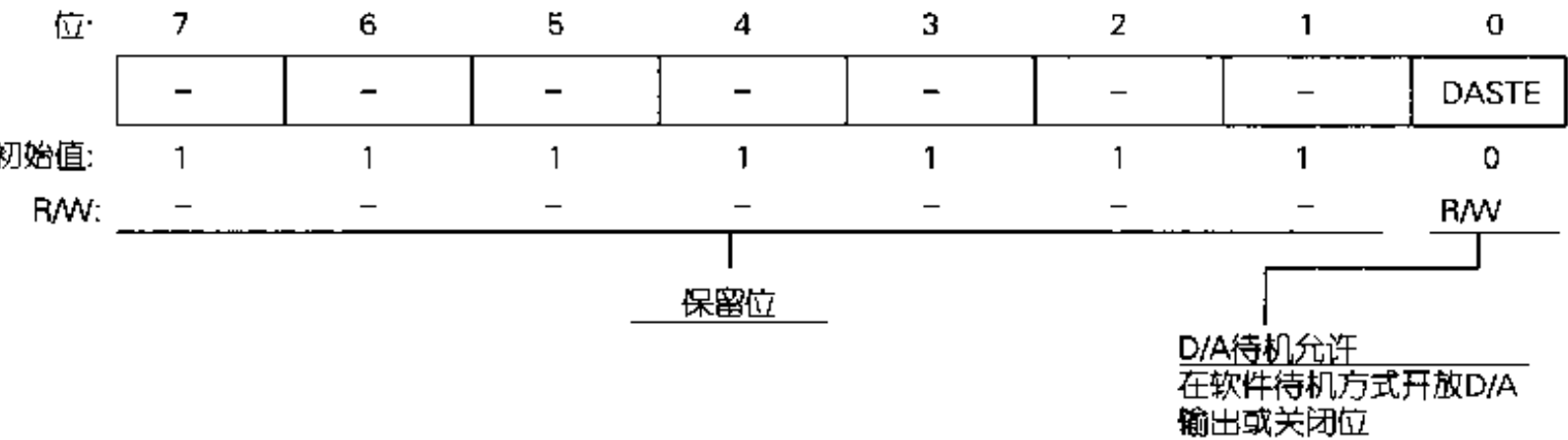
表 3.11 D/A 转换器寄存器的构成

地址*	名 称	简 称	R/W	初始值
H'FFDC	D/A 数据寄存器 0	DADR0	R/W	H'00
H'FFDD	D/A 数据寄存器 1	DADR1	R/W	H'00
H'FFDE	D/A 控制寄存器	DACR	R/W	H'1F
H'FFSC	D/A 待机控制寄存器	DASTCR	R/W	H'FE

* 表示地址的低位 16 位



• 图 3.12 DACR 寄存器



• 图 3.13 DASTCR 寄存器的详细说明

表 3.12 DA 转换器的控制

位 7	位 6	位 5	说 明
DAOE1	DAOE0	DAE	
0	0	—	关闭通道 0、1 的 D/A 转换
	1	0	开放通道 0 的 D/A 转换
		1	关闭通道 1 的 D/A 转换
	1	1	开放通道 0、1 的 D/A 转换
1	1	0	关闭通道 0 的 D/A 转换
		1	开放通道 1 的 D/A 转换
	1	1	开放通道 0、1 的 D/A 转换
	1	—	开放通道 0、1 的 D/A 转换

38

3.5 中 断

在相扑机器人竞赛中,要设计出能一边检测摔跤场的外区,一边追踪对手,同时还能进行武器控制等,同时进行多种控制的程序设计,实际上,这是相当困难的事情。在第 10 章中介绍的相扑机器人安装了上臂,进行这个臂的控制,要使用间隔定时中断处理,主体部分的处理,是通过分别控制实现的。换句话说,在中断处理中,仅仅考虑臂的控制程序设计就可以,这样主程序设计就可以一心一意地进行对手或摔跤场的检测。

就像一个人在上班时,虽然在工作,但是总想着快到下班时间了,不时地看表,工作是不会有进展的。计算机也是一样,只注意时间,其它的处理就草率。与此相比,如果有一个一到某一时刻就会立即通知你的表该有多好。做这种事的就叫定时中断。即计算机进行寻找对手作战等工作时,在某一定时间间隔从定时器如果发出信号:“到时间了”,计算机就进行臂的控制,其控制一结束,又回到原来的工作。如果把 这个控制时间定为 5ms 左右,主要工作

和控制臂的处理时间处理均衡,对手逃不掉,臂也流畅地动作下去。

下面,就有关具体地中断方法进行叙述。对于中断,可能从各种各样

CPU 功能发生。为此,根据各功能处理程序的内容不同,发生中断时,由于移动执行各处理程序,所以将存储器作为中断向量设定,如表 3.13 所示。

表 3.13 中断向量

中 断 源	发生源部件	向量号码	向量地址	IPR	优先级
NMI	外部引脚	7	H'001C~H'001F		高 ↑
IRQ0		12	H'0030~H'0033	IPRA7	
IRQ1		13	H'0034~H'0037	IPRA6	
IRQ2		14	H'0038~H'003B	IPRA5	
IRQ3		15	H'003C~H'003F		
IRQ4		16	H'0040~H'0043	IPRA4	
IRQ5		17	H'0044~H'0047		
保留	—	18	H'0048~H'004B		低 ↓
		19	H'004C~H'004F		
WOVI(间隔定时器)	监视定时器	20	H'0050~H'0053	IPRA3	
CM1(比较匹配)	刷新控制器	21	H'0054~H'0057		
保留	—	22	H'0058~H'005B		
		23	H'005C~H'005F		
IMIA0(比较匹配/输入捕捉 A0)	ITU 通道 0	24	H'0060~H'0063	IPRA2	
IMIB0(比较匹配/输入捕捉 B0)		25	H'0064~H'0067		
OV10(溢出 0)		26	H'0068~H'006B		
保留	—	27	H'006C~H'006F		
IMIA1(比较匹配/输入捕捉 A1)	ITU 通道 1	28	H'0070~H'0073	IPRA1	
IMIB1(比较匹配/输入捕捉 B1)		29	H'0074~H'0077		
OV11(溢出 1)		30	H'0078~H'007B		
保留	—	31	H'007C~H'007F		
IMIA2(比较匹配/输入捕捉 A2)	ITU 通道 2	32	H'0080~H'0083	IPRA0	
IMIB2(比较匹配/输入捕捉 B2)		33	H'0084~H'0087		
OV12(溢出 2)		34	H'0088~H'008B		
保留	—	35	H'008C~H'008F		
IMIA3(比较匹配/输入捕捉 A3)	ITU 通道 3	36	H'0090~H'0093	IPRB7	
IMIB3(比较匹配/输入捕捉 B3)		37	H'0094~H'0097		
OV13(溢出 3)		38	H'0098~H'009B		
保留	—	39	H'009C~H'009F		
IMIA4(比较匹配/输入捕捉 A4)	ITU 通道 4	40	H'00A0~H'00A3	IPRB6	
IMIB4(比较匹配/输入捕捉 B4)		41	H'00A4~H'00A7		
OV14(溢出 4)		42	H'00A8~H'00AB		
保留	—	43	H'00AC~H'00AF		

在 H8-3048F 单片机中,作为中断源,外部中断源有 7 个(NMI、IRQ 0~

5),根据来自片内周边模块的要求有 30 个内部中断源。对于所要求的中断

源片内周边模块,有监视定时器(WDT)、刷新控制器、16 位定时集成单位(ITU)、DMA 控制器(DMAC)、串行通信端口(SCI)以及 A/D 转换器等。如表 3.13 所示,中断向量地址各中断源可分别分配。

其中,使用监视定时器(WDT)的间隔定时器功能,使其发生中断。发生这个中断时,在向量号码 20(H'0050~H'0053)上预先写入地址跳变,执行其地址上书写的指令。在这个地址写好控制臂的程序。

3.5.1 监视定时器

H8-3048F 单片机具有监视定时器(WDT),这是为了对原来系统状态进行监视的定时器,然而,它向同时具有作为间隔定时器的功能,所以使用它可以发生中断。

定时计数器(TCNT)每出现溢出(计数值超过设定值),间隔定时器就

发生中断。发生这种中断时,就进行臂的控制处理。

间隔定时器可以选择以下 8 个计数器输入时钟脉冲 $\phi/2$ 、 $\phi/32$ 、 $\phi/64$ 、 $\phi/128$ 、 $\phi/256$ 、 $\phi/512$ 、 $\phi/2048$ 、 $\phi/4096$ 。

这时,满计数(可以计 0~255 数值)时间分别为:0.032ms、0.512ms、1.024ms、2.48ms、4.96ms、8.192ms、32.768ms,以及 65.536ms。

3.5.2 定时计数器

定时计数器(TCNT)可以是 8 位读写递增计数器。

TCNT 的值溢出时(H'FF 进一步递增计数到 1,瞬间返回 H'00),TCSR 寄存器的 OVF 位被置 1。其中,由于发生中断,所以读取溢出标志,在 TCNT 上写入 5ms 部分计数值。在此 5ms 以后,还可能发生中断(参见图 3.14)。

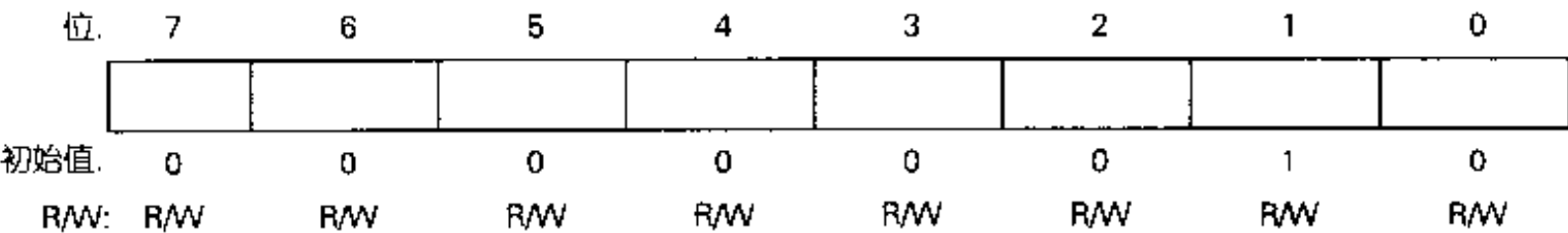


图 3.14 定时计数器 TCNT

3.5.3 定时控制/状态寄存器

定时控制状态寄存器(TCSR)是可以进行 8 位读写的寄存器,其功能是选择输入定时计数器的时钟脉冲和模式。

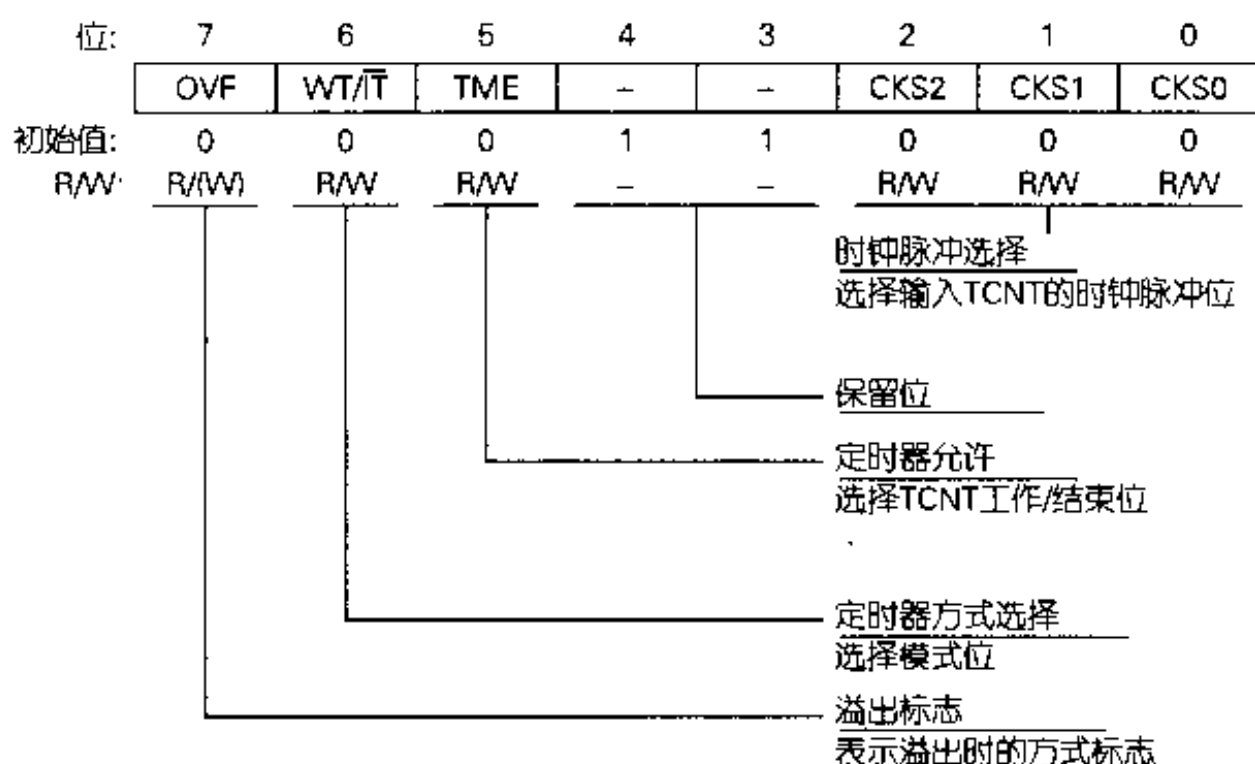
选择定时器模式位设定为 0 时,就变成间隔定时器方式。将定时器允许位设定为 1 时,监视定时器(WDT)开始计数。通过用选择时钟脉冲的 CKS2~CKS0 位选择内部时钟脉冲,决定递增计数时间。选择 $\phi/512$ 计数

器输入时钟脉冲,产生 5ms 的间隔定时器时的 OKS 位指定为 011。

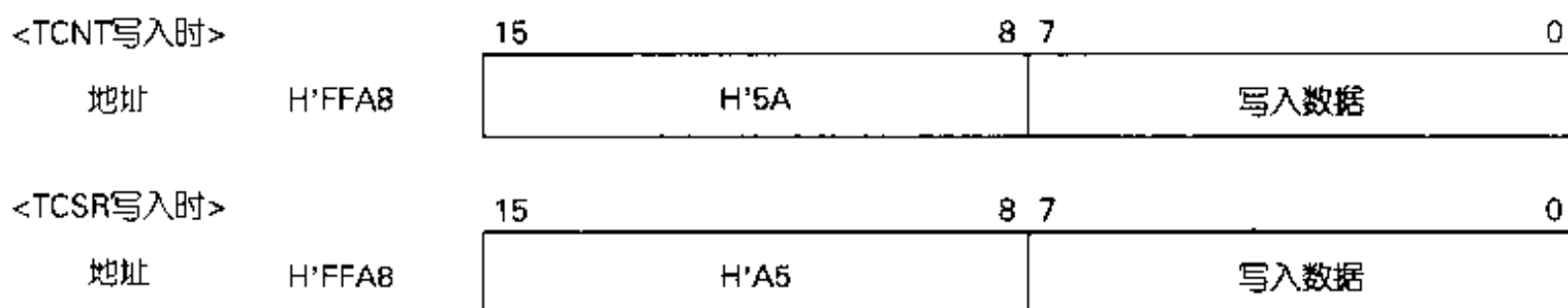
3.5.4 TCNT、TCSR 的写入

进行 TCNT、TCSR 这两个寄存器写入时,必需以特殊的方式进行。由于写入时的地址与 TCNT、TCSR 变为

同一地址,所以写入 TCNT、TCSR 时,写入低位字节的数据将高位字节 H'5A(TCNT 时)或 H'A5(TCSR 时)作为 1 个字(16 位)数据写入。据此,低位字节的数据写入 TCNT 或 TCSR。这由硬件决定,所以必须仅限于使用这个 CPU(参见图 3.16)。



• 图 3.15 TCSR

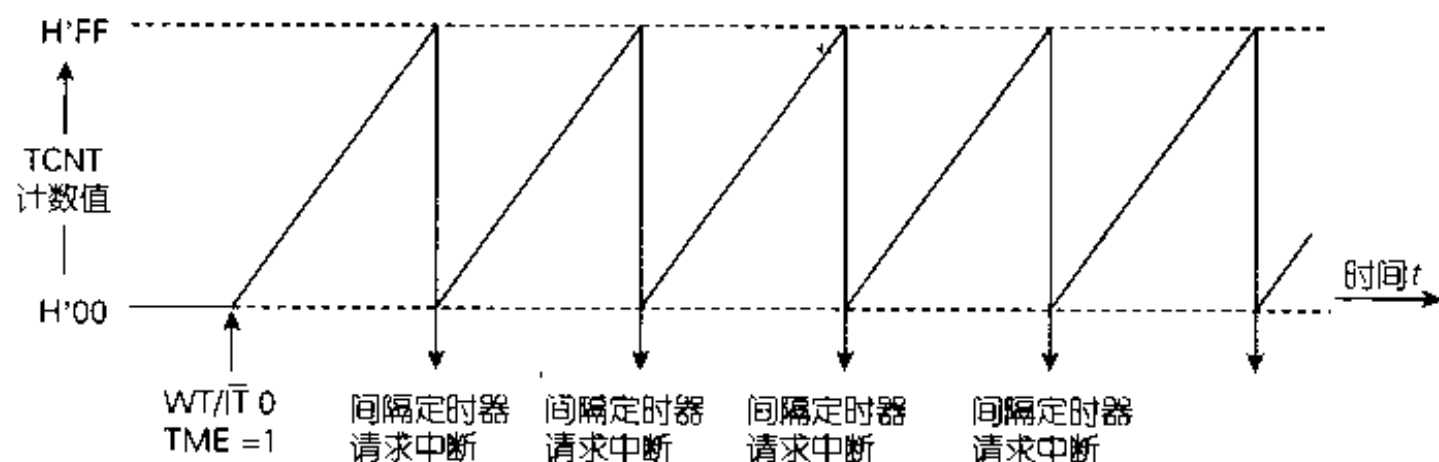


• 图 3.16 TCNT、TCSR 的写入

3.5.5 间隔定时器的工作原理

图 3.17 表示间隔定时器的工作原理。作为 5ms 的间隔定时器使用

时,将 TCSR 寄存器的 WT/IT 位清 0,将 TME 位置 1。其中,将 H'00111011 设在 TCSR 寄存器中。



• 图 3.17 间隔定时器的工作原理

作为间隔定时器工作时, TCNT 每次溢出时, 间隔定时器发出中断请求。这里, 发生中断后, 在中断处理程序中

再次将 TCNT 的值设定为 100。由此, 再经过 5ms 时, 还可以发生中断(如图 3.17 所示)。

3.6 充分利用串行通信 R232C 端口

在普通个人电脑的背面, 作为串行通信出口附有 RS232C 的 D-SUB9 连接引脚(最近正在更换 USB 或 IEEE1394 等, 对笔记本个人电脑这些遗留装置大多数被排除)。试使用这个 RS232C 端口和机器人进行通信。

在 H8 中, 因为使用串行口进行片内快速存储器(Flash ROM)的改写, 所以使用相同口进行通信。

3.6.1 串行通信

在串行通信中, 有时钟脉冲同步式或异步式通信。这里讲述的是异步式通信。异步式通信方式如图 3.18 所示, 用开始位和停止位通过插入的数据可以进行发送、接收数据。开始位意味着通信开始, 停止位意味着通

信终止。用这些位组合异步通信, 就是用很少的信号线互相进行读入、取出数据。

在计算机间进行读入、取出数据时, 当然决定于通信速度(1 秒钟传送多少位), 但是这些收、发数据位数或停止位等彼此通信的数据不组合好, 就不能正确地进行通信。

3.6.2 串行通信端口

首先简单叙述一下 H8-3048F 单片机内置的串行通信端口(SCI)。

表 3.14 中表示 SCI 上的寄存器, 通过使用这些寄存器, 设定通信方式或通信条件, 可以进行计算机之间的通信。在 H8-3048F 上有 2 个通信口。为向 H8-3048F 片内的快速(Flash)

ROM上写入程序,使用通道 1。这里即使使用相同的通道 1,寄存器通道 0 也要进行与通道 1 完全相同的准备,所以仅仅变更地址,0 通道也同样可以处理。

3.6.3 串行方式寄存器

串行方式寄存器(SMR)是进行串行通信端口(SCI)的串行通信格式的设定及为了选择波特率发送器的时钟脉冲源的 8 位寄存器。

位 7 设定通信模式,0 为异步模

式,1 为同步模式。

位 6 如图 3.19 所示,表示发送、接收数据的位长。CHR = 0,则数据长 8 位;若 CHR = 1,则数据长为 7 位。

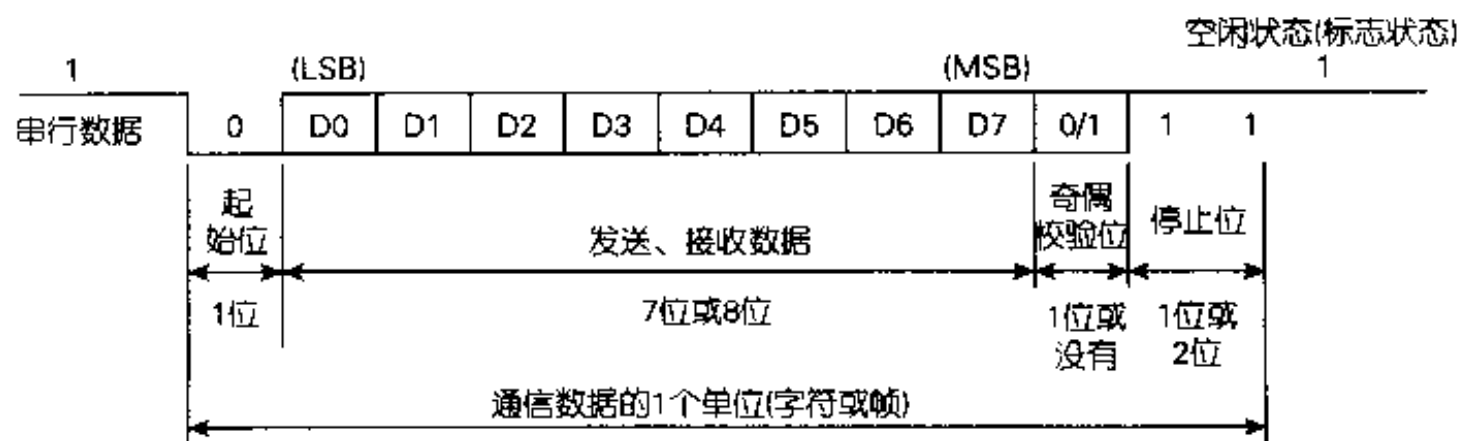
位 5 为奇偶校验的设定位,位 3 设定停止位的长度。为 0 时则为 8 位。

位 1、0 为时钟脉冲选择,详见后述。

这里,异步通信方式设定 8 位数据长,没有奇偶校验位,1 为停止位,时钟脉冲设定没有分频。这时的 SMR 寄存器上设定值变为 0x00。

表 3.14 SCI 寄存器的构成

通道	地址	名称	简称	RW	初始值
0	H'FFB0	串行方式寄存器	SMR	R/W	H'00
	H'FFB1	位率寄存器	BRR	R/W	H'FF
	H'FFB2	串行控制寄存器	SCR	R/W	H'00
	H'FFB3	发送数据寄存器	TDR	R/W	H'FF
	H'FFB4	串行状态寄存器	SSR	R/W	H'84
	H'FFB5	接收数据寄存器	RDR	R	H'00
1	H'FFB8	串行方式寄存器	SMR	R/W	H'00
	H'FFB9	位率寄存器	BRR	R/W	H'FF
	H'FFBA	串行控制寄存器	SCR	R/W	H'00
	H'FFBB	发送数据寄存器	TDR	R/W	H'FF
	H'FFBC	串行状态寄存器	SSR	R/W	H'84
	H'FFBD	接收数据寄存器	RDR	R	H'00



• 图 3.18 异步通信方式位

3.6.4 串行控制寄存器

串行控制寄存器(SCR)为设定串行通信端口的发送和接收工作,进行请求中断的开放/关闭等的寄存器。另外,也用于选择时钟脉冲源。

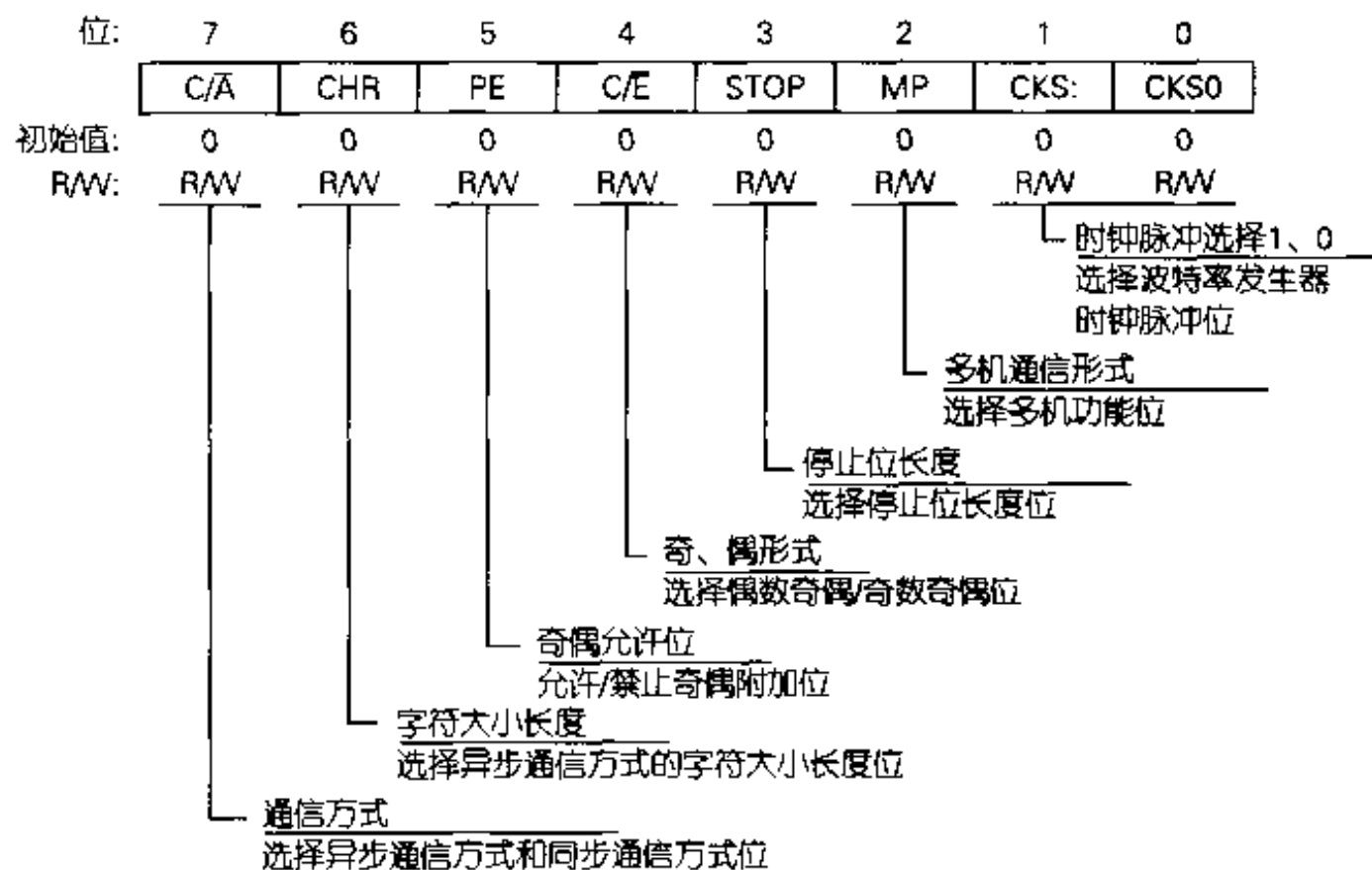
位 7、6、3、2 分别为设定中断允许

其中,SCR 寄存器上设定 0x30(参见图 3.20 所示)。

/禁止位,1 为允许,0 为禁止。

位 5、4 为允许/禁止发送、接收工作位,1 为允许,0 为禁止。

位 1、0 为选择时钟脉冲源位。时钟脉冲源作为内部时钟脉冲,没有使用中断,所以最好只允许发送、接收。



• 图 3.19 SMR 寄存器的详细说明

3.6.5 串行状态寄存器

串行状态寄存器(SSR)是 8 位寄存器,主要表示 SCI 的工作状态。是一种为了了解发送、接收是否结束或是否出现错误的寄存器。

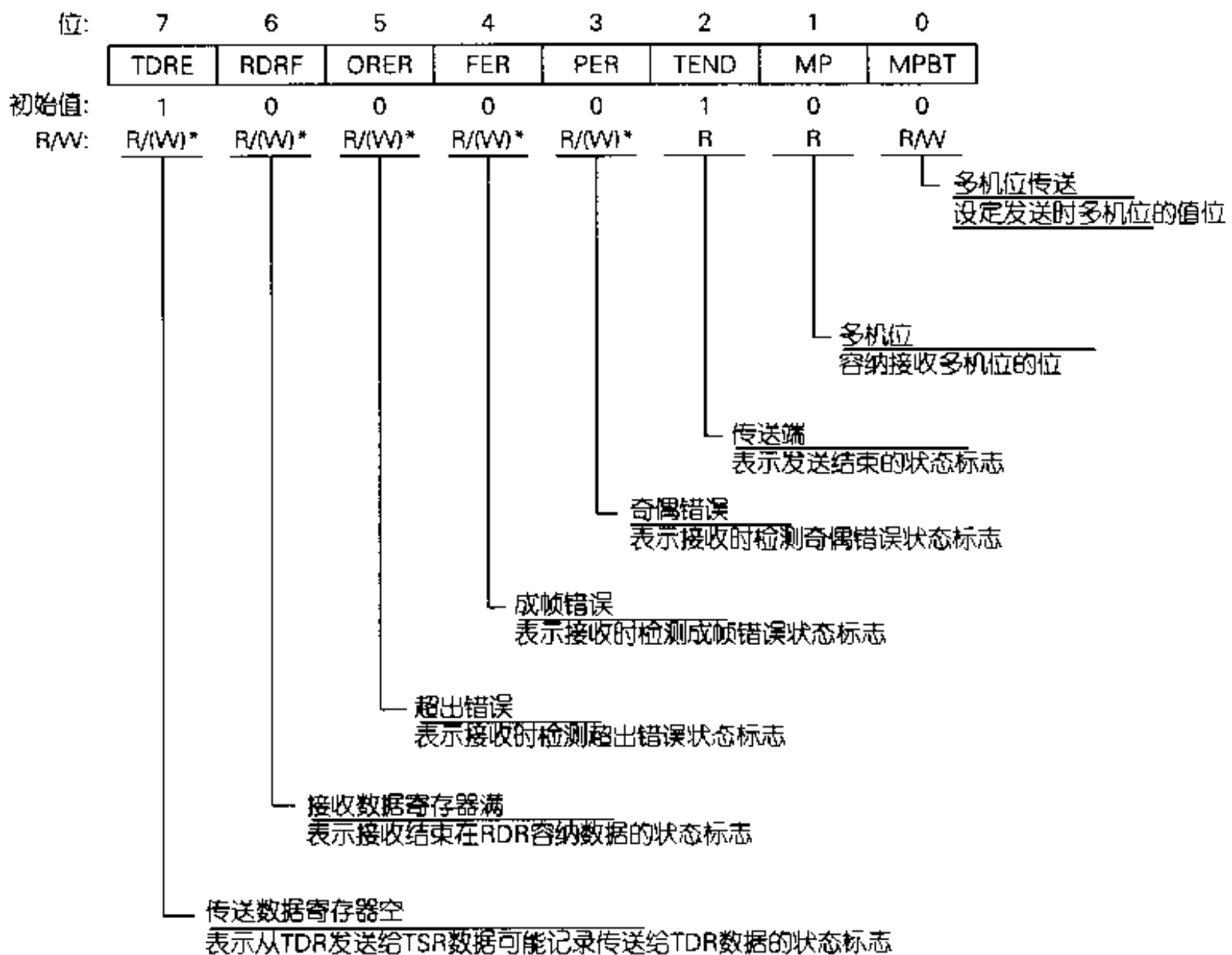
位 7 可以根据程序将数据写入发送数据寄存器(TDR),这个数据和转

送数据传输给 TSR 寄存器。这里为了向 TSR 上按顺序串行传送数据,即使将数据写入 TDR 寄存器上就行,这时,这位就变为 1。

位 6 接收结束,数据写入 RDR,所以读取很方便,这时变为 1。以这 2 位为主一边进行检测,一边进行通信(参见图 3.21 所示)。



• 图 3.20 SCR 寄存器的详细说明



• 图 3.21 SSR 寄存器的详细说明

3.6.6 位率寄存器

位率寄存器(BRR)为用 SMR 的 CKS1、CKS0 位和选择波特率发生器的工作时钟脉冲比较,设定串行发送/接收波特率(通信速度)的 8 位寄存器。

就是将用以下的计算公式求得的值设定在 BRR 上,以决定通信速度。
异步通信方式:

$$N = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$$

式中, B 为位率(bit/s),
 N 为波特率发生器 BRR 的设定值($0 \leq N \leq 255$),
 ϕ 为系统时钟频率(MHz),
 n 为波特率发生器输入时钟脉冲

位($n = 0, 1, 2, 3$)。

n 和时钟脉冲的关系从表 3.15 中选定。

表 3.15 通信速度的计算

n	时钟脉冲	SMR 的设定值	
		CKS1	CKS0
0	ϕ	0	0
1	$\phi/4$		1
2	$\phi/16$	1	0
3	$\phi/64$		1

这个 CPU 的时钟脉冲是 16MHz, SMR 的设定,因分别把 CKS1, CKS0 设定 0,所以 n 为 0。另外,将波特率设为 9600bit/s(在 1 秒钟传送 9 600 位)时, n 变为 51。如果将这个值输入 BRR 寄存器,波特率设定结束。同样,对其它的波特率,也在实例程序中事先定义。

chapter

4

微型计算机插件板

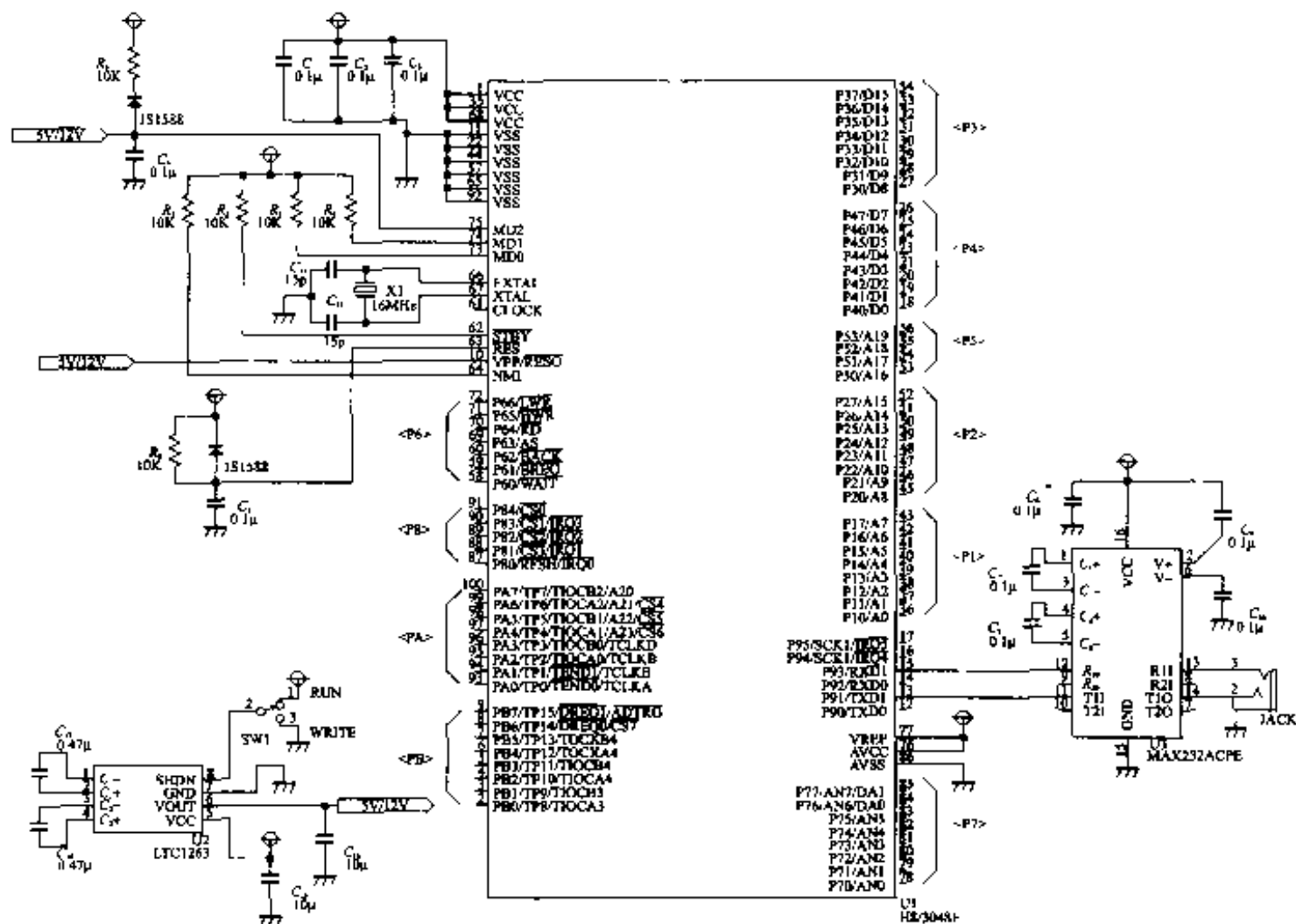
本章是对H8微型计算机焊接技术的挑战。其中，由于预料到制作中的困难，所以介绍了市场上出售的微型计算机插件板。另外，也对只读存储器写入器（ROM WRITER）的制作方法进行讲解。

4.1 制作 H8 微型计算机插件板

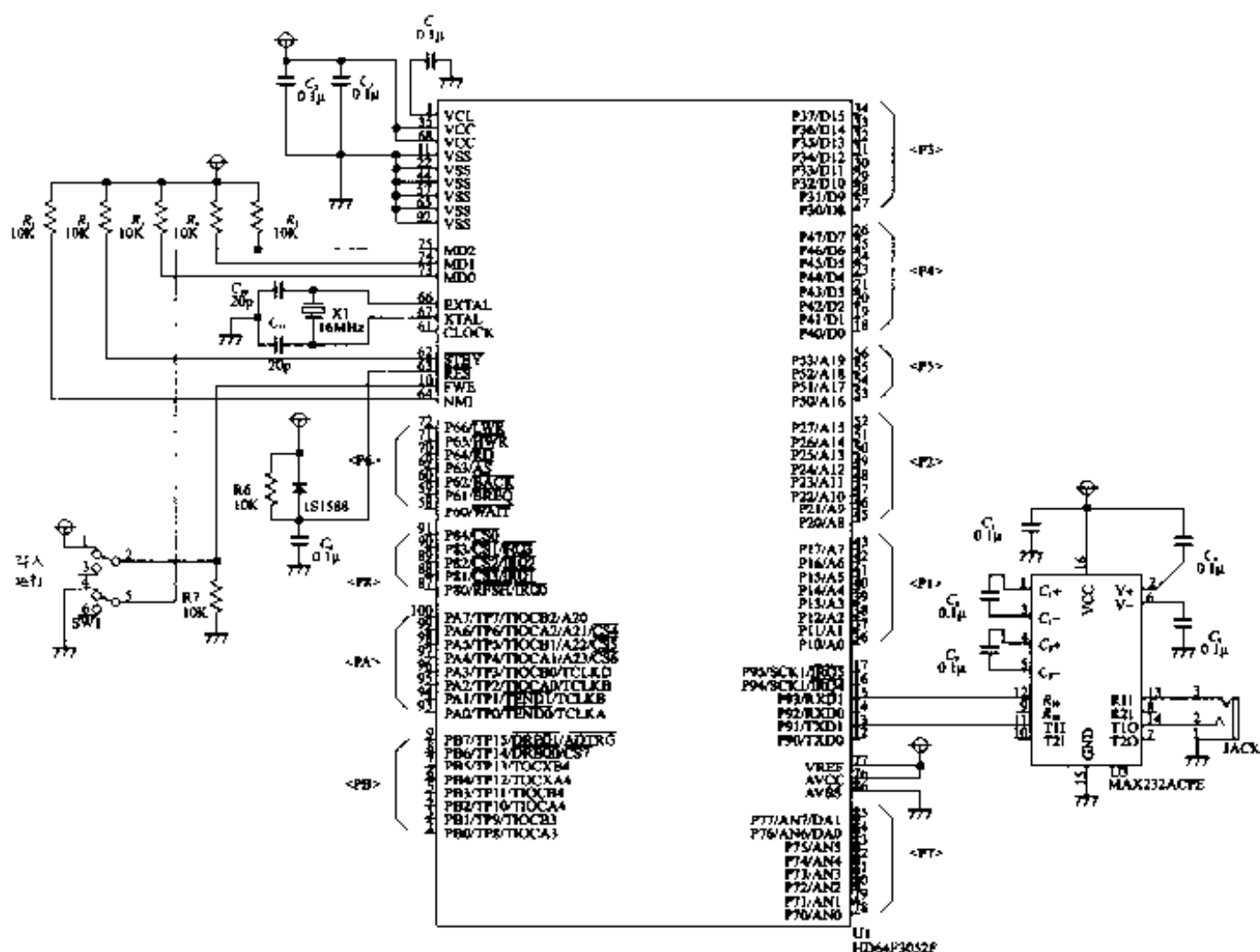
近年来,很多 IC 模块越来越小型化,DIP 型模块越来越少了。因此,可以认为,如果不能把焊接技术和组装技术有机组合,就不能制作微型计算机插件板。H8 系列微型计算机也不例外,大部分 H8 系列都是扁平封装组件,芯片固定在基片上,形状标准化。所以使用市场上出售的转换基片等,变换间距,就可以装配在通用基片上。

图 4.1 是 H8-3048F 用插件板电路图,图 4.2 是 H8-3052F 用的插件板

电路图。刚刚问世的 H8-3052F 一般还没有普及,而 H8-3048F,12V 的片内快速存储器(Flash ROM)的写入电压变为 5V,片内存储器也比 3048F 快速存储器(Flash ROM)增加 4 倍,达 512k 字节,RAM 也增加 2 倍,到 8k 字节。最大时钟脉冲也稍微提高,可以工作到 18MHz。本书中所述的使用内部 I/O 的各种功能和 3048F100% 兼容,可能是平移过来的。电路也比 3048F 版改进了。



• 图 4.1 H8/3048F 微型计算机插件板电路图



• 图 4.2 H8/3052F 微型计算机插件板电路图

4.1.1 工具准备

做电子技术工作多少要有些工作经验,以前使用过的工具大体上可用,然而,对于焊接扁平封装的 IC 模块,焊接技术水平必须高。

• 电烙铁(30~50W)

不管电烙铁是哪一种形式,要考虑到进行微细工作的情况,事先注意烙铁尖上细小的焊锡。另外,电烙铁瓦数过小时,低于元件和基片的热容量焊锡不能流畅地流出,所以电烙铁瓦数多少还是大一点的好。

• 树脂芯焊丝

考虑进行精密的焊接作业情况,如果有 0.5mm 直径的树脂芯焊丝最方便。

• 焊剂

在基片上焊接扁平组件时,事先将焊剂涂敷在基片上,它有在铜箔图形以外焊锡很难流过的效果。焊剂有液体状或糊剂状的,要选择使用方便的焊剂。

• 焊锡吸收线

焊锡吸收线为难编织铜线,它吸收不要的焊锡很方便。也可以采用吸引形焊锡吸收器,但由于是用吸引力吸收焊锡,往往用于将不需要部分的焊锡或蔓延到基片线路图形上的焊锡

全部吸收。所以对于精密作业,最好采用焊锡吸收线。

• 镊子

由于需要一边夹紧精密部件,一边按紧它焊接,所以有镊子最方便。

• 放大镜

即使是扁平组件,但当间距为0.5 mm 时,用肉眼很难检测,一定事先注意准备放大镜。

4.1.2 电路图和元器件清单

对 H8 的 RAM 或 ROM,因为是内置存储器装置,所以如果仅做单一工作变成非常简单的电路。构成大型电路,则为时钟脉冲源、RS232C 通信及快速写入电源电路(仅 H8 /3048F)(如表 4.1)。

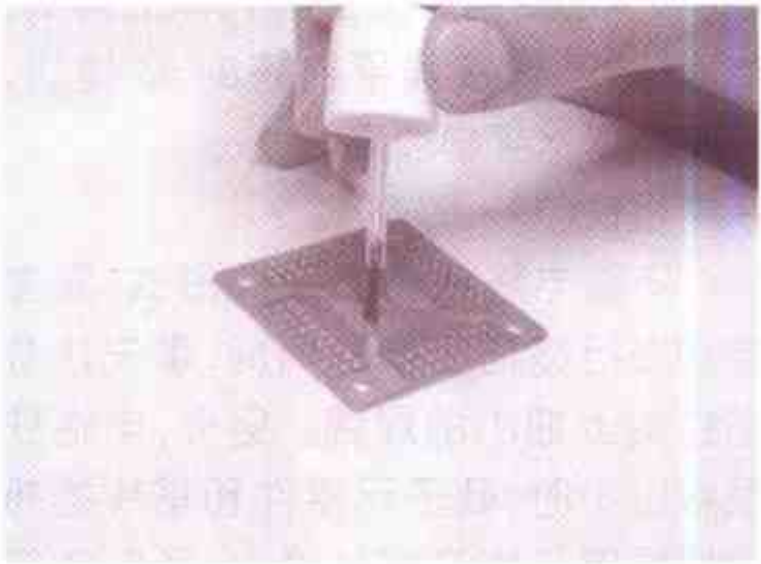
表 4.1 元器件清单

元器件名称	H8 /3048F 插件板	H8 /3052F 插件板
转换基片	Sanhayat QFP 间距转换基片 QFP-53 形	
普通基片	0.1 小间距 100mm×150mm	
H8 CPU	HD64F3048F18(U1)	HD64F3052TF18(U2)
DC/DC 转换器(IC)	LTC1263(U2)	—
RS232C 基准转换器(IC)	MAX232ACPE(U3)	
晶体信号传输频率(X1)	16MHz(X1)	
电阻	10kΩ(R ₁ ~R ₅)	10kΩ(R ₁ ~R ₇)
陶瓷电容器	0.1μF50V(C ₁ ~C ₁₀)	0.1μF50V(C ₁ ~C ₉)
	15pF(C ₁₁ ,C ₁₂)	15pF(C ₁₀ ,C ₁₁)
	0.47μF50V(C ₁₃ ,C ₁₄)	—
钽电解电容器	10μF16V(C ₁₅ ,C ₁₆)	—
转换开关	(SW1)	
φ3.5 硬插座	(JACK)	
小信号二极管	1S1588(D ₁ ,D ₂)	1S1588(D ₁)

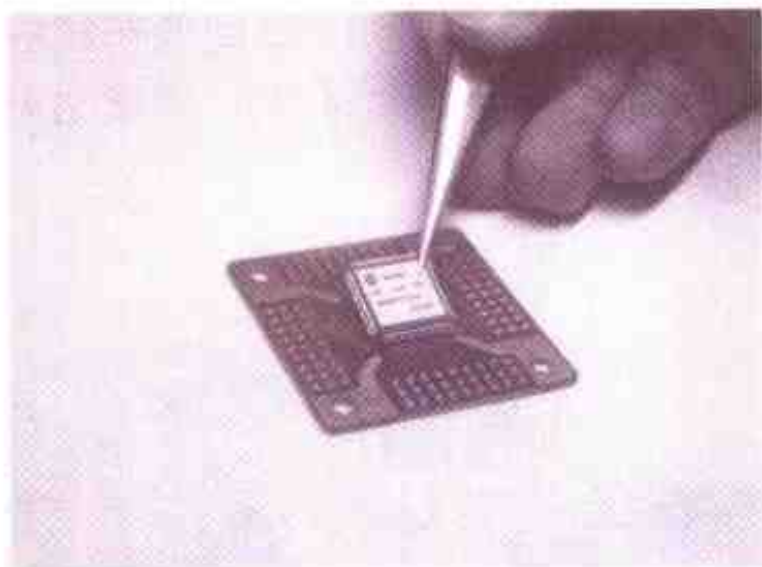
4.1.3 实装作业要点

首先,进行最重要的 CPU 焊接。如照片 4.1 所示,在转换基片的线路图形上涂上焊剂,一涂就可以了。

将 IC 模块对准基片的图形非常仔细地放好。这时要注意不要把 IC 模块的引脚弄弯(参见照片 4.2)。

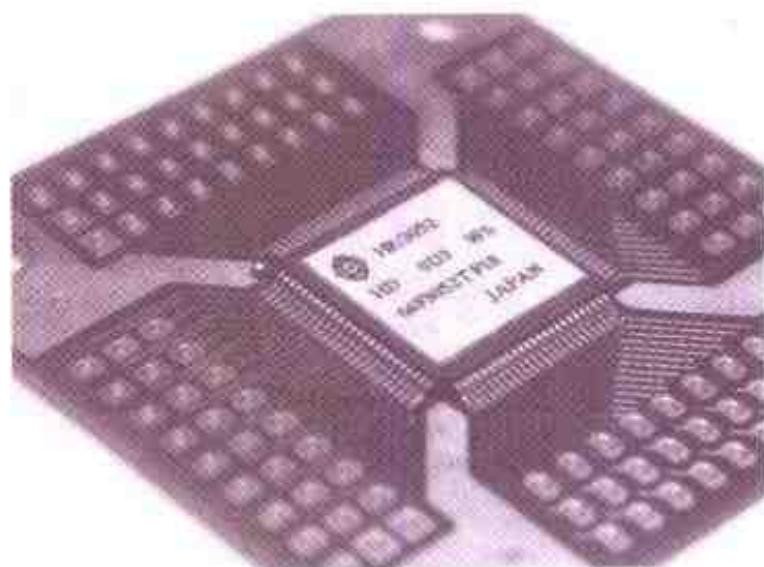


• 照片 4.1 涂焊剂



• 照片 4.2 将 H8 芯片放在基片上

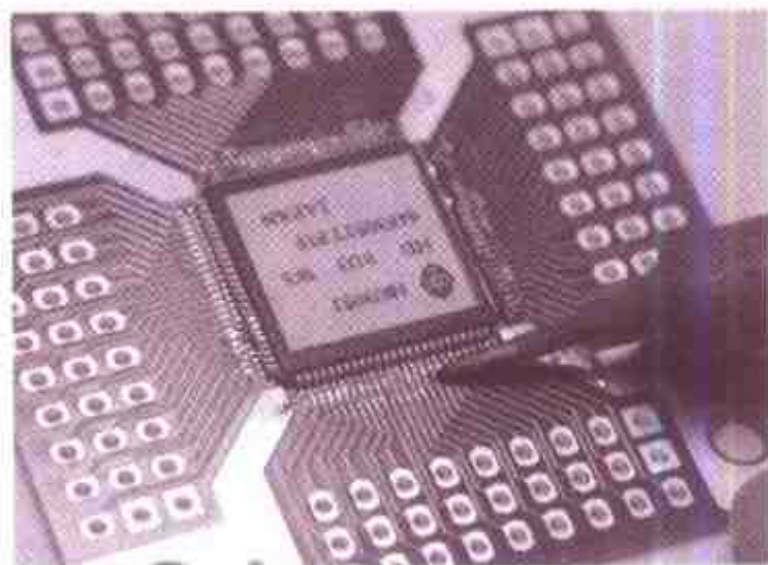
IC 模块不要偏移于基片,手要一边用力按住,一边将四角的引脚焊好(参见照片 4.3)。电烙铁上少沾一些焊锡进行焊接。要慢慢小心地作业。只要四角完全焊好了,模块就不会偏移了。



• 照片 4.3 在四角引脚上进行焊接

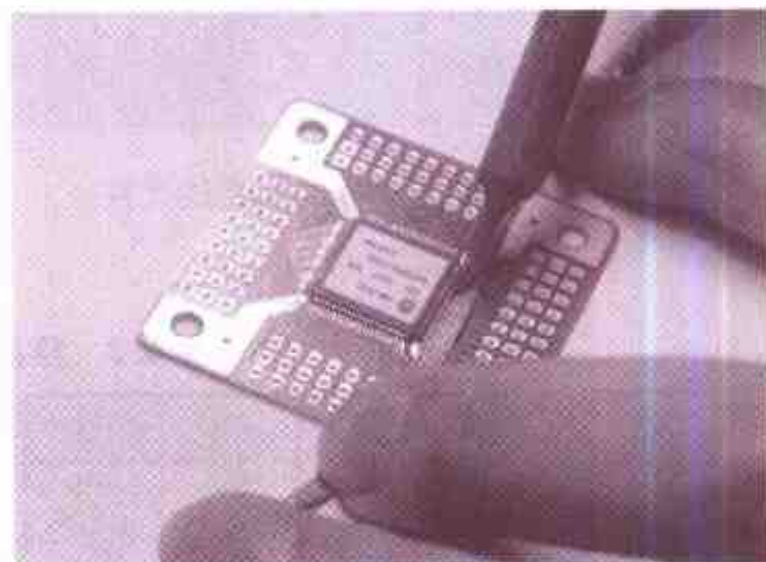
用同样的要领,在电烙铁上少沾一些焊锡,将 IC 模块各边的引脚和基片线路图形搭接好,像用笔写字一样,一边将基片和引脚加热,一边慢慢移动电烙铁(参见照片 4.4)。不要中途停止,一口气焊完,这样,焊点变得均匀牢固。电烙铁上沾锡过量时,容易把相邻元件的引线连接在一起,所以

希望稍微试一试。如果焊锡流到不要的部位也没关系,因为最后还要进行修正作业,所以先不必介意,继续焊接好其它边。



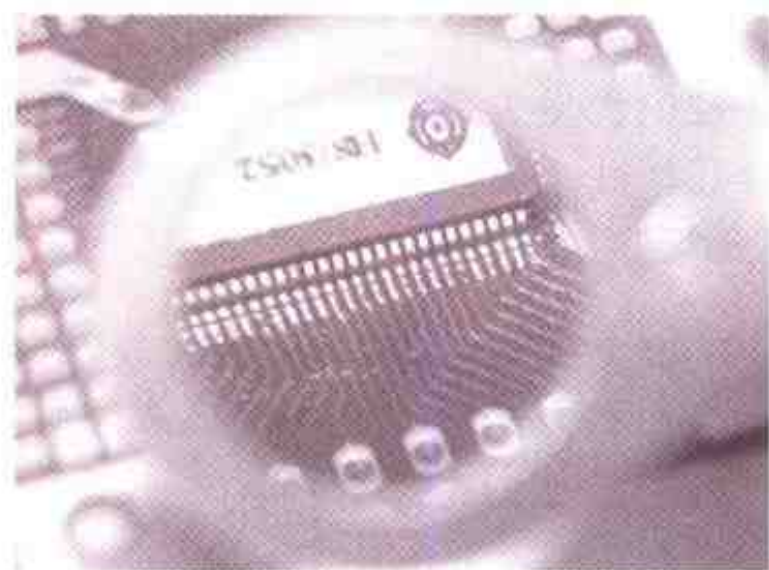
• 照片 4.4 进行引脚焊接

如果焊锡完全把相邻引线连接起来了,要进行修正,此时,先在完全连接的部分少涂些焊剂。备好焊锡吸收线,将其放在完全连接的部分,将烙铁放于其上,焊锡被加热并被吸收线吸收(参见照片 4.5)。



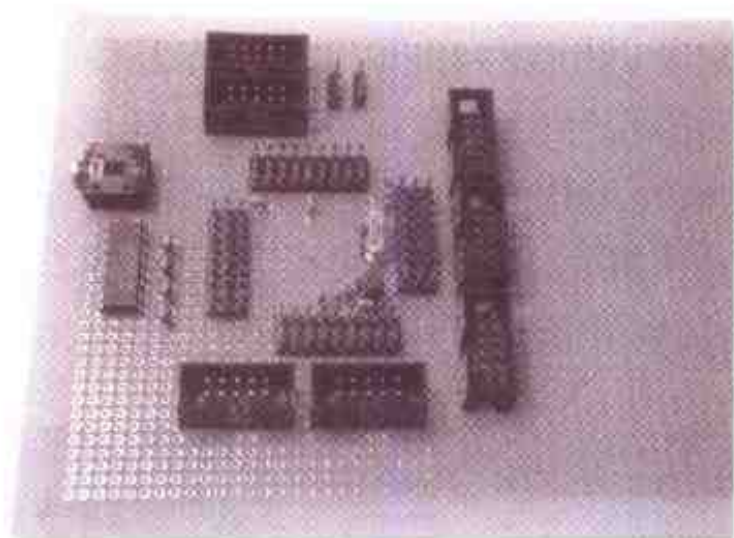
• 照片 4.5 吸收焊锡

焊锡流入 IC 模块的引脚和基片线路图形相接触的部分即可,所以要用放大镜检验焊口是否有虚焊(参见照片 4.6)。



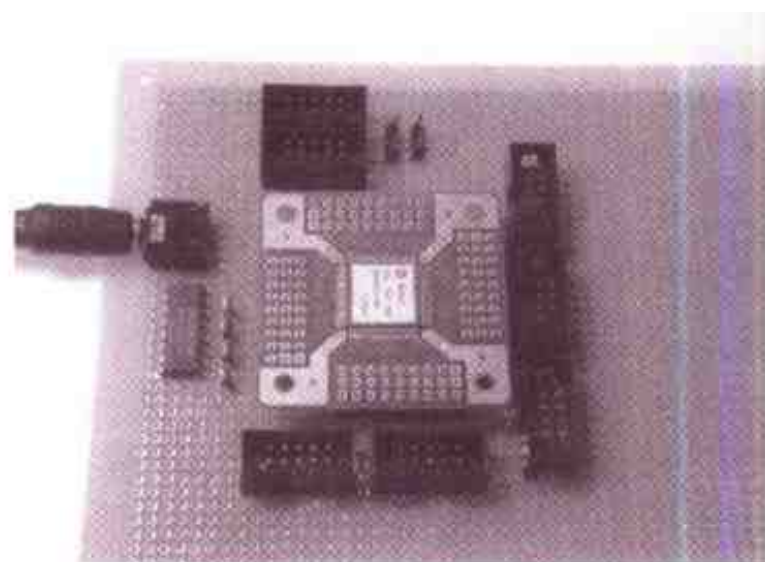
• 照片 4.6 用放大镜检验焊口

到此,就已经焊接完了。再将其
它的元件在通用基片上配置、布线,经
由插接器装配转换基片。这些在
电路图上没有表示,作为时钟脉冲源



• 照片 4.7 在普通基片上配置的元件

的晶体振荡元件与 CPU 间的布线一
定尽量短(参见照片 4.7)。完成品见
照片 4.8。



• 照片 4.8 完成品

4.1.4 接插件输出

接插件的连接如表 4.2 所示。8
位各端口的输出,用 10 条扁平导线用的
接插件(P1~PB)随着电源一起引
出。另外,用 3 条顶部引线(JP1~
JP2)作为 PWM 输出,对 A/D 输入、
D/A 输出的检验用引线则另设。

表 4.2 接插件规格明细表

使用这些输入/输出接插件,各个功能的程序将在下章以后介绍。

4.1.5 确定工作状态

在布线检验结束之前,一定注意不要供电。一旦出现错误,此前的工作将完全白费。布线检验,按照焊接结束的顺序,用标识器等一边进行检验,一边涂去焊剂即可。

如果检验结束,接上电源试试看。为避免发生问题,先将手指放在 CPU 上,一边确认温度是否急剧上升,一边

供给电源。如果出现出乎意料的情况,马上将供给电源切断。不易买到的元器件中最重要的是 CPU。作者在执笔的时候只买到 H8/3052F 以前样品的鉴定版产品。其它不易买到的元器件,在元器件工厂等如果预定,即使是邮购有时也可以搞到。

另外,图 4.1 中使用 LT1263 快速存储器的写入电压供给 12V,而这个 IC 模块由于是扁平封装,所以与 H8 说明书 40mA 相比,电源容量多少有些不足,也可以使用 MAX6626ACPA。

4.2 市售微型计算机插件板的选定

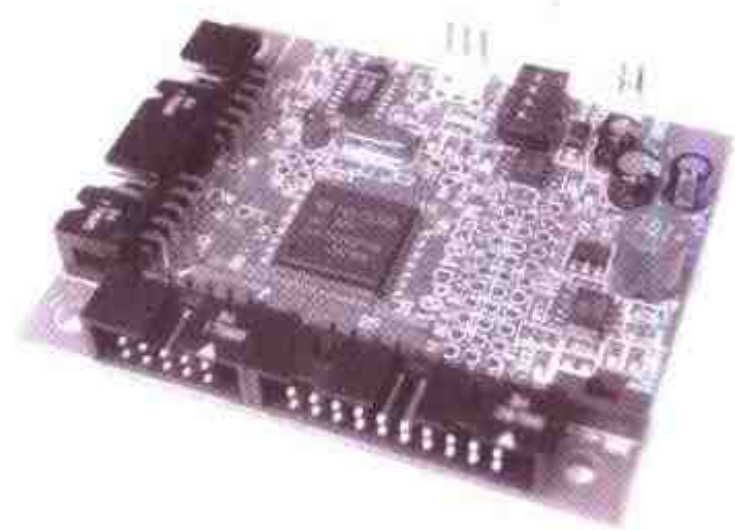
对市售的搭载 H8/3048F 的微型计算机插件板进行了调查。表 4.3 为微型计算机插件板及开发工具的比较表。此外,最近很多厂家都在出售插件板。用低价格购入开发环境完备的部件,在今后开发机器人方面是很重要的。

为了开发面向初学者的使用 C 语言的机器人,建议使用价格更低的“秋月电子”的微型计算机插件板。“黄页软件”的开发环境完成度高,即使中级以上用户也能使用这些软件,所以在此作概要介绍。另外,日立纯正开发工具或免费软件 GCC 有各方面的优点。本书中介绍使用免费软件 GCC 的 C 语言程序。

1. 北斗电子的 H8 微型计算机插件板

在日本微型计算机圣杯机器人竞

赛中指定的插件板是北斗电子制作的。对大会参加者,供给插件板、汇编程序和 C 编译程序。用接插件将快速存储器(Flash ROM)写入用的电源电路装载在插件板上,所以仅使用 5V 电源和连接 RS232C 电缆就能开发程序(参见照片 4.9)。



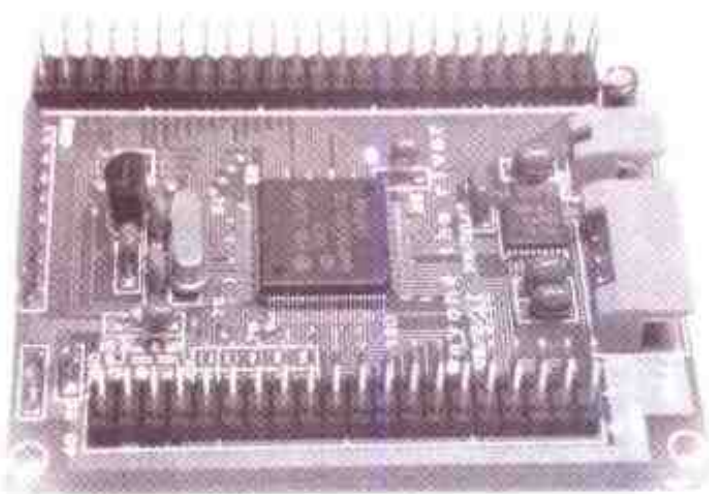
• 照片 4.9 北斗电子的 CPU 插件板

表 4.3 微型计算机插件板及开发工具的比较表

	北斗电子	秋月电子	黄页软件	Cygnus 解法 (仅开发软件)
微型计算机插件板	配发给日本计算机 圣杯机器人竞赛参 加者,参加费 ¥10 000 日元	AKI-H8 微型计算机 插件板 ¥3 800 日 元 AKI-H8 配套元件 (含 AKI-H8) ¥7 800 日元	开 发 配 套 软 件 ¥38 000 日元	使用自己制作的或左 栏所述的插件板
ROM 写入器电路	装载微型计算机插 件板	装载母模插件板	调试程序单	使用自己制作的或左 栏所述的电路
ROM 写入器软件	可载	附带	附带	使用自己制作的或左 栏所述的软件
汇编程序、连接程序	附带	附带	附带	附带
C 编译程序	附带	另外出售(日立制作) ¥2 000 日元	附带	附带
外部存储器	附带(日立制作) 没有	没有	在调试程序单上装 载 ROM・RAM	—
特征	日本微型计算机圣 杯机器人竞赛用	价格低	整理开发环境	—

2. 利用黄页软件的 H8/3048F 开发配
套软件

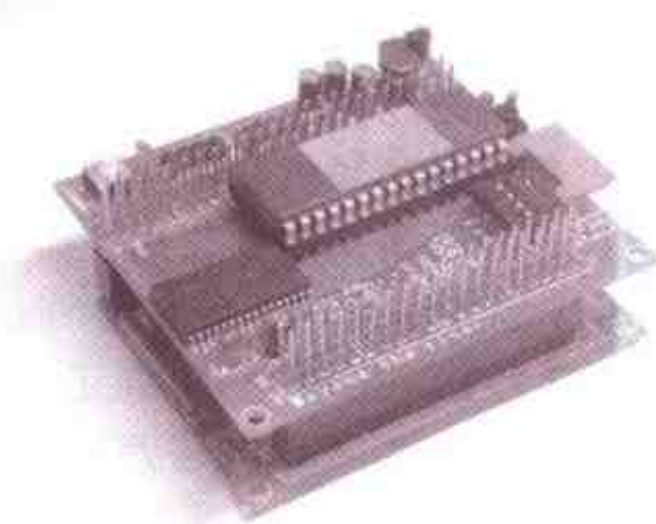
在这个开发配套软件中包含汇编
程序、C 编译程序及各种工具, H8/
3048F 的 CPU 插件板(参见照片 4. 10)
及调试插件板。



• 照片 4. 10 H8/3048F 的 CPU 插件板

调试插件板和 CPU 插件板如照

片 4. 11 所示,使用 2 层重叠结构。调
试插件板和 CPU 插件板如此频繁地
拔出和插入,这样的开发不可能实现。
将快速存储器(Flash ROM)写入器事
先装配在调试插件板上,用 5V 单一电
源也可以开发。



• 照片 4. 11 开发中的状态

统一开发环境(程序编辑、编译程

序、调试等可由一个程序进行)事先准备好,在 Windows 基本环境中可以利用。画面 4.1 为用统一开发环境程序开发中的画面。



• 画面 4.1 黄页软件统一开发环境

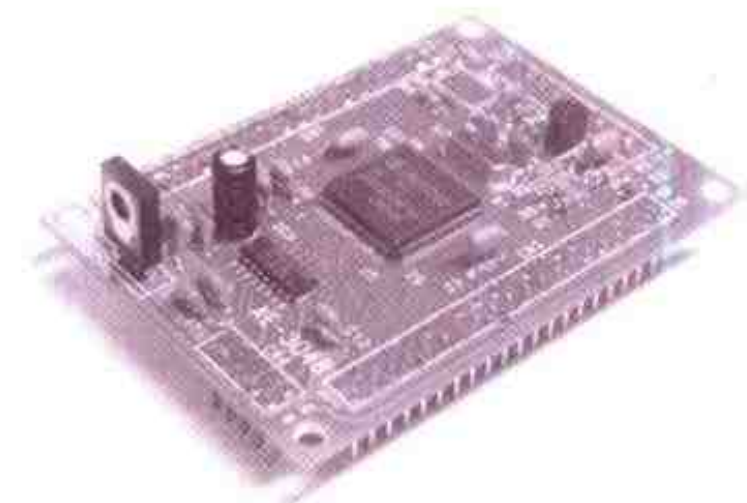
另外,通信软件也事先编入,使用 C 语言标准输出 I/O(Printf 等),从 H8 微型计算机插件板的通信口可以简单地在个人电脑上输出文字数据,对调试是非常有效的。

3. AKI-H8 微型计算机插件板

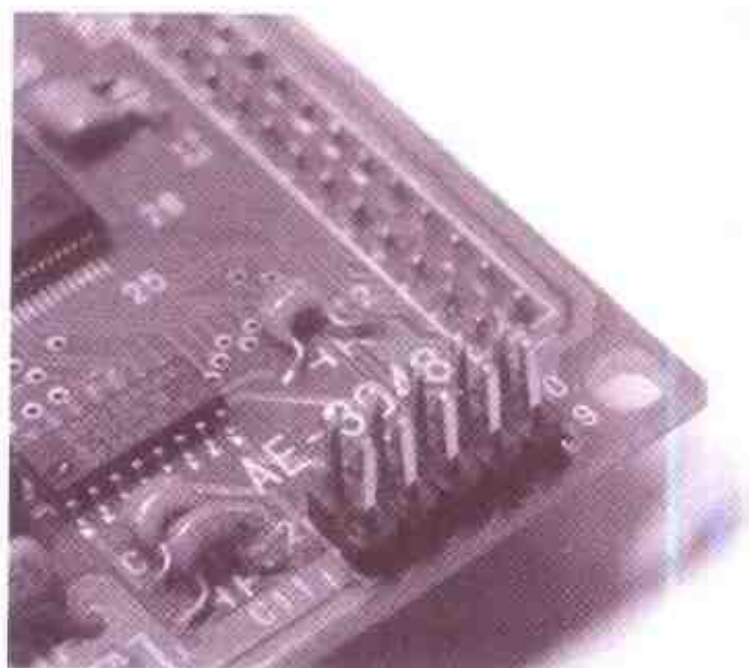
AKI-H8 微型计算机插件板为自制配套元件,所以今后制作机器人的工作中还是经常要用到电烙铁的。扁平封装的部件均为实装件,所以只要焊接上一部分半导体器件、电阻和电容器就可以完成(参见照片 4.12)。

将这种微型计算机插件板的接插件 CN4,如照片 4.13 所示,如果朝上

装配,则后面叙述的快速存储器(Flash ROM)写入器插件板也可以使用,所以没有使用包括 AKI-H8 开发配套元件的母模插件板,向装载在 AKI-H8 上的 H8 快速存储器(Flash ROM)的写入,用 5V 单一电源就可以进行工作。



• 照片 4.12 完成的 AKI-H8 插件板

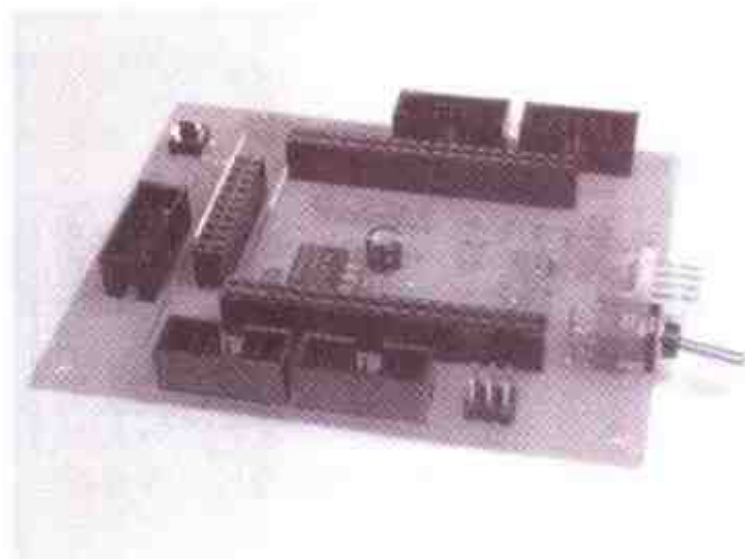


• 照片 4.13 将快速存储器(Flash ROM)写入器用接插件朝上装配(CN4)

4.3 AKI-H8 用 I/O 插件板的制作

由于在 AKI-H8 微型计算机插件板上不可能安装直接输入/输出装置,所以准备 I/O 插件板(参见照片 4.14 所示)。就是仅仅安装简单的电源和接插件。接插件如表 4.2 所示,均事先备好。

为了获得 CPU 驱动电压的 5V 稳压电源,要装载 5V 的 3 引脚调节器。5V 电源电路可使用第 9 章的图 9.7 所示电路等。



• 照片 4.14 完成的 I/O 插件板

4.4 制作快速存储器写入器插件板

对附带 AKI-H8 开发配套元件的母模插件板,从外部供给 15V 左右的电源,用插件板上的 12V 调节器读出快速写入电压 12V,供给 AKI-H8。其它公司大部分的插件板用 5V 单一电源对应写入,所以 AKI-H8 大致也和它们同样,希望事先准备好同样的环境。

下面介绍快速存储器(Flash ROM)写入器插件板,首先介绍电路图上装载的 DC/DC 转换器部分及将通信插孔装在 AKI-H8 用的其它的基片上(参见图 4.3 所示)。

在 AKI-H8 的 CN4 上汇集 5V 电源,快速存储器电源,串行通信联系端子,所以将其元件面朝上,就是为了考虑在 AKI-H8 上用“龟仔”状态连接快速存储器(Flash ROM)写入器插件板

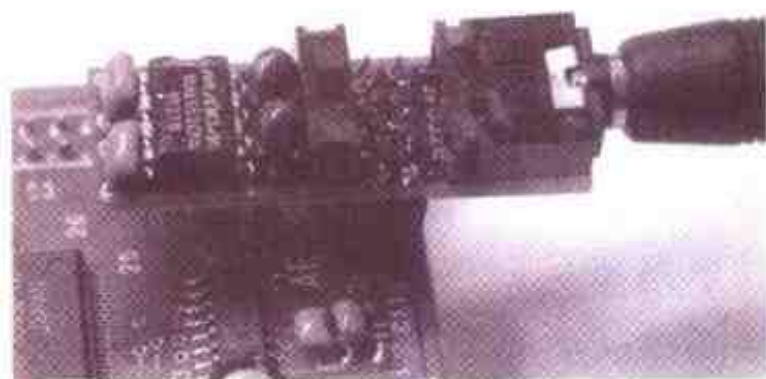
(参见照片 4.15)。



• 照片 4.15 AKI-H8 用快速存储器(Flash ROM)写入器插件板

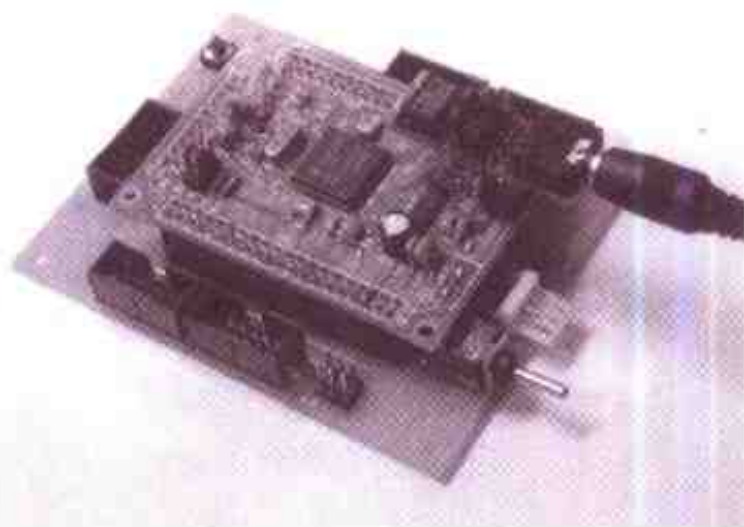
为了和个人电脑通信的端子连接,将其变成拨出插入方便的低频用的管脚插孔,而立体管脚插头正好有

发送(TX),接收(RX)和接地 3 个端子,所以这种人工通信接插件用起来很方便(参见照片 4.16)。

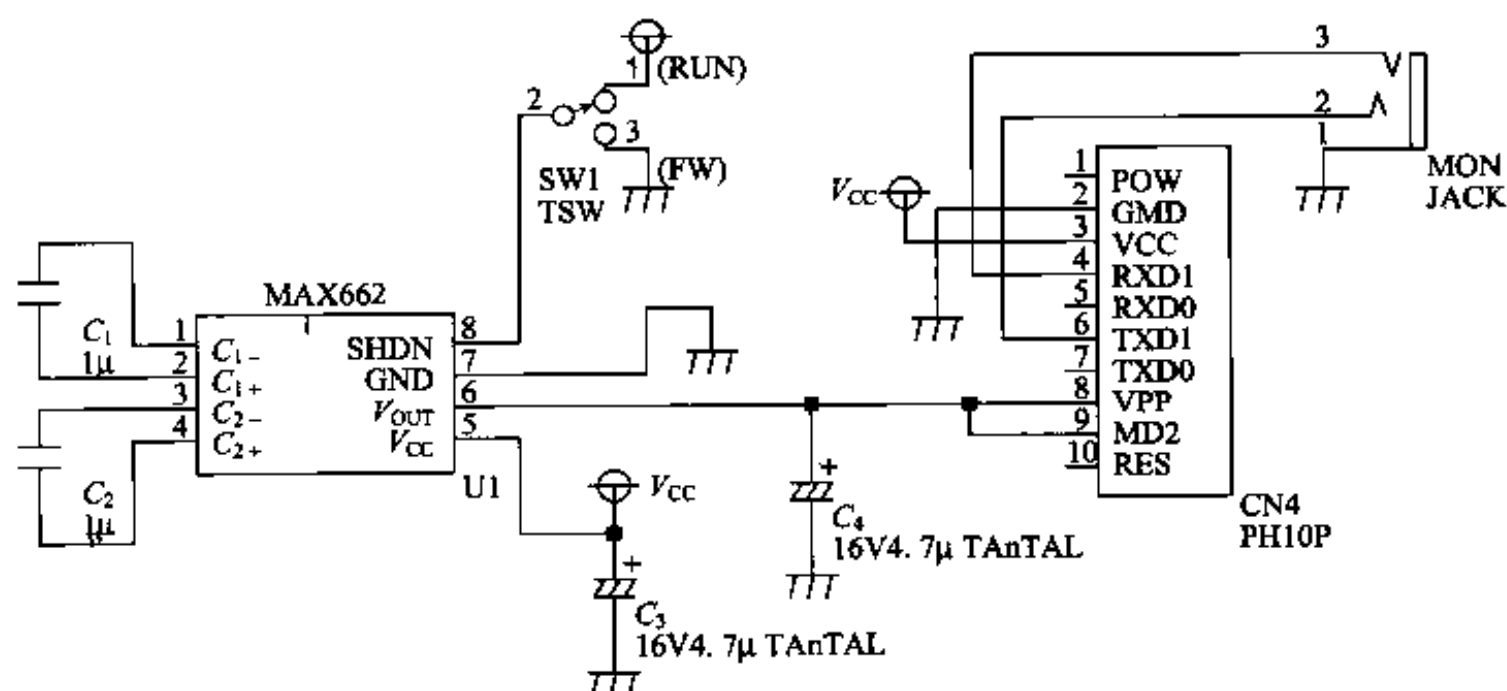


• 照片 4.16 在 AKI-H8 上带有实装通信电缆的状态

这里从外部供给电源,仅仅用 5V 对应写入快速存储器。不需要写入快速存储器时,即使将 ROM 写入器插件板完全卸下也不介意(参见照片 4.17)。



• 照片 4.17 和 I/O 插件板连接



• 图 4.3 快速存储器(Flash ROM)写入器电路图

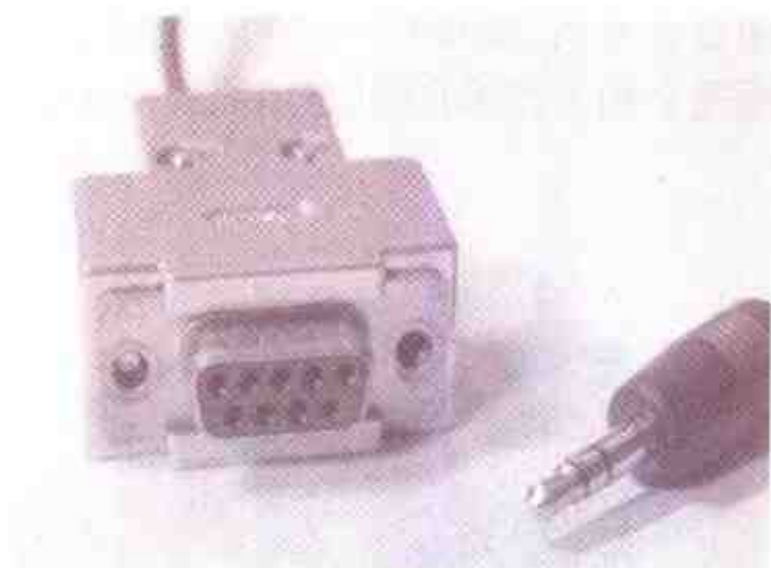
4.4.1 通信电缆的制作

为了将装载 AKI-H8 上的快速存储器(Flash ROM)写入器插件和个人电脑连接,需要制作通信电缆

(RS232C 电缆)。用个人电脑将作成的程序写入微型计算机,也要通过这个电缆进行。

PC/AT 互换个人电脑的 RS232C 接插件如照片 4.18 所示那样的形状。

另外,电缆的连接图如图 4.4 所示。



• 照片 4.18 PC/AT 互换机的 RS232C 接插件和立体管脚插头

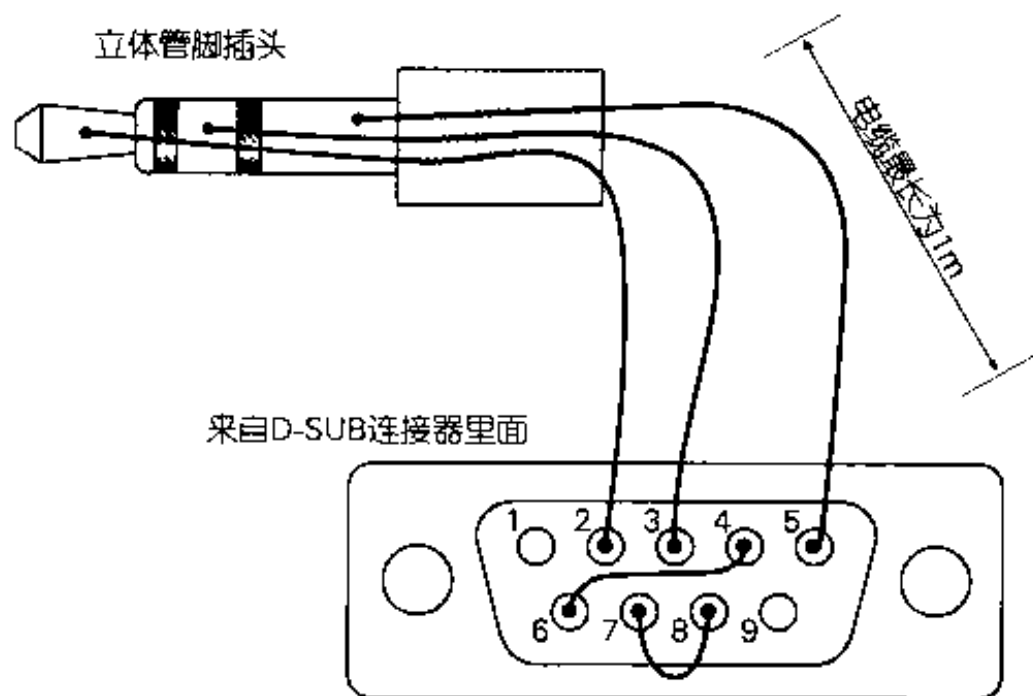
4.4.2 H8 快速存储器传送程序

在个人电脑上如果开发 H8 的工作程序,将制作的程序从个人电脑上

传送不到 H8 就没有意义了。各个公司也准备了原型传送工具。但是,这次在使用 GNUPro Toolkit 时,没有附带传送工具。仅仅为了传送不可能购买另外的开发工具,所以这次将自己喜欢的工具加在一起自己制作。

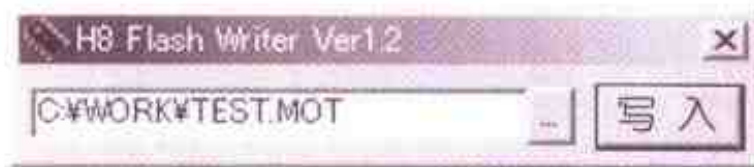
向 H8 上传送程序的方法的详细说明这里省略,而使用 C 编译程序等, H8 的程序开发结束,最终作完文件的格式,正如大家所知道的那样,各个公司是各不相同的。这里的 H8 的快速存储器(Flash ROM)传送程序能与全部相对应。

扩充端子 MOT 的情况,可对应摩托罗拉 S 格式文件(对应日立制作编译程序、黄页软件制作的编译程序及 GCC)判断,将 BIN 的情况作为二进制文件(对应 GCC)处理。

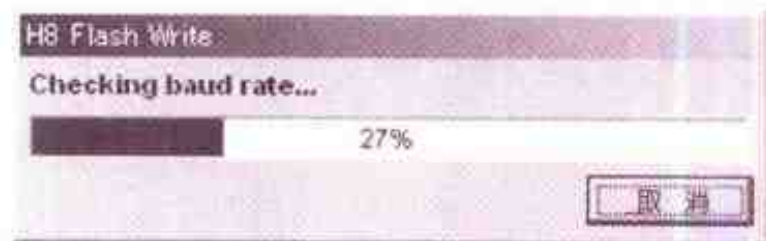


• 图 4.4 RS232C 信号接线图

该程序在 Windows95/98/NT 环境下可以利用,所以希望读者一定参照上述说明使用。



- 画面 4.2 FW.EXE 起动时的画面



- 画面 4.3 向 H8 传送中的画面

chapter

5

微型计算机的 开发环境

本章中将介绍日立制作所开发的H8微型计算机的软件开发环境。最近，机器人开发中主要使用C语言。因此本章在介绍使用免费的编译程序GCC构筑的开发环境的同时，还将对ROM WRITER电路和写入用软件进行讲解。

5.1 必要的开发环境

在自律型机器人开发中,除必需有个人电脑外,还必需有编辑程序、汇编程序和 C 编译程序。

1. 编辑程序

所谓编辑程序就是用来制作、编辑源程序的,换句话说,就是如字处理机(wordpro)的功能。在 Windows 中,设置了用标准清单的文本编辑程序,然而也有市场上出售的商品 WZ 编辑程序及共享软件的“秀丸”编辑程序等各种各样的软件。读者可以试试看,只要是适合于自己使用的软件即可。

2. 汇编程序

所谓汇编程序就是根据每个微型计算机的不同,用最小指令形式组成的汇编语言记述的程序、用连接程序可以处理转换文件的一种方法。

本书中以介绍 C 语言程序为主,所以大体上没有使用汇编语言。但是,即使用 C 语言编制程序,也必需变成所谓初始化程序、用进行初始化处理的汇编语言记述的程序。初始化程序依赖于其微型计算机的硬件部分或仅仅描述需要 C 编译程序的初始化处理,所以若一次准备好,以后大致没有必要涉及到它。

但是,为了使其微型计算机工作,汇编语言可以用它的最小单位记述指令,所以,有时在高速化应用中往往记述一部分指令。而且,在一般情况下,

往往将中断循环处理的向量地址记述在初始化程序上,然而对 GCC,因为是用特殊的方法实现,所以没有这种必要。

3. C 编译程序

C 编译程序是一种变换工具,它把用 C 语言记述的程序通过连接变换成可处理的文件。在机器人程序开发过程中,关于 C 语言只要记住以下指令即可。这些指令通过有关 C 语言的基础读物中即可了解。

C 语言与汇编语言相比,可以说是会话式的程序语言。

前处理程序

#include

#define

控制语句

if else while

switch case

for

break

变量的型式

unsigned char

int

算术运算符

+, -, *, /, %

关系运算符

<, <=, >, >=, ==, !=

逻辑运算符

!, &&, ||

增量、减量运算符

++ , --

位运算符

~, <<, >>, &, |, ^

4. 连接程序

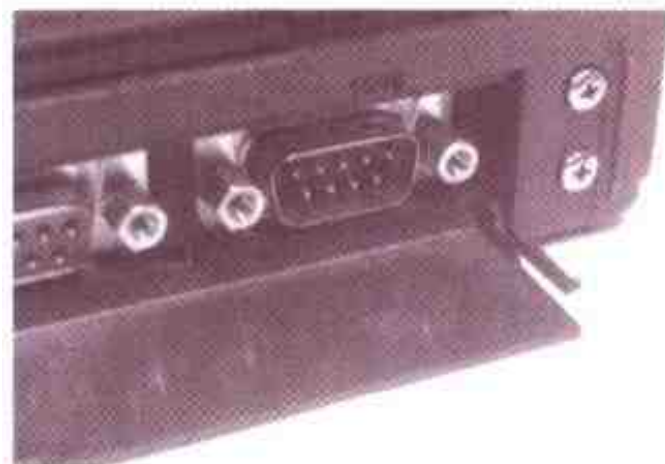
连接程序是用前述的汇编程序或 C 编译程序将作成的文件归纳成一个程序,转换成最终的目标程序的一种

方法。

对于本书中使用的 GNUPro Toolkit, C 编译程序、汇编程序及连接程序等没有各自的操作方法,而是采用称作 GCC 的批处理方法。为此,没有将上述方法作为直接目标,而用 GCC 处理时往往具有选择等意思,所以希望事先具有大体的印象。

5.2 C 语言开发环境的准备

首先,准备好个人电脑,以下均以具有个人电脑 DOS/V 作为前提开展讨论。没有附带 RS232C 端口时,不能向 H8 微型计算机传送程序,所以希望一定使用装配 RS232C 端口的微型计算机。并且,以 Windows95 或 Windows98、Windows2000 中任何一种的 OS 作为备用(参照照片 5.1 所示)。



• 照片 5.1 RS232C 通信接口

5.3 使用 GCC 编制 H8 微型计算机程序

1. GCC(参见画面 5.1)

听说过 GCC 这个词吗? GCC 是 Free Software Foundation, Inc 的 GNU 计划(project)的免费软件的一部分(用户运行软件、复制、散发、研究、变更,并且有改良的自由)。GNU 基本将叫做 Linux 的 OS 上工作的软件公开,源代码也公开。

这里的免费软件享有:

• 不问目的,均有运行程序的自由(第

0 个自由)。

- 研究程序如何工作,在这个程序上,使用者有根据需要加以修正、采用的自由(第 1 个自由)。能够得到源代码是它的前提条件。
- 能够帮助身边的人,有将其复制再散发的自由(第 2 个自由)。
- 改进程序,若团体公用都能受益,有将你改进点向大家发表的自由(第 3 个自由)。能够得到源代码在这里

是前提条件。

GNU 计划的网址如下:

<http://www.gnu.org/home.ja.html>

并且, GCC 是 GNU Compiler Collection(GNUCC)的简称。它指能够在各种 CPU(H8/300 或 SH 系列·ARM·SPARC·M68k 等)上可以输出工作代码的 C 编译程序(实际上汇编程序或连接程序等也需要)。



• 画面 5.1 <http://www.gnu.org/software/gcc/gcc.html>

2. GNUPro Toolkit 的安装

GCC 在 Linux(如 UNIX 的 OS)上利用的时候很多。现在,对于在 PC 微机的 Windows 上, GCC 不能简单工作。收录在 CD-ROM 中的 GCC 是 Cygnus 解析公司提供的、称为[Cygnus GNUPro Toolkit]的程序包(参见画面 5.2),开发 GCC 初始程序需要一些工具,使它能在 Windows 上工作。这些在日立 AMERICA 网址上:

<http://semiconductor.hitachi.com/repository/tools/Gnutool.html>

与将 GCC 作为第 3 种方法公开的相同。



• 画面 5.2 Cygnus(GNU) 编译程序

这个程序包可以从上述网址下载,然而由于它约有 90M 字节,所以下载时要花费相当的时间。(参见画面 5.3 所示)。

File Transfer Progress									
2024-07-27 08:00:00 - 2024-07-27 08:00:00									
File Name Size Progress Status Speed Time Left Error Count Retry Count Total Size Total Time									
binutils	400	100%	完成	10000000	0:00:00	0	0	400	0:00:00
gcc	175,440	100%	完成	10000000	0:00:00	0	0	175,440	0:00:00
libgcc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libstdc++	400	100%	完成	10000000	0:00:00	0	0	400	0:00:00
libg++	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_32	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_ppc	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_s390	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_z	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_x86_64	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_arm	20,340	100%	完成	10000000	0:00:00	0	0	20,340	0:00:00
libg++_mips	20,340								

上,将所谓的[setup.exe]备用程序由探测器等实行一下看(参见画面 5.4 所示)。



• 画面 5.4 CYGNUS

安装流程如下:

(1) Wellcome

原封不动地按[Next]键。

(2) Software license Agreement

如果同意特许文件按[Yes]键。

(3) Information

是 Cygnus Solutions 公司的信息,原封不动按[Next]键。

(4) Choose Destination Folder

指定安装用文件夹,最好使用默认的安装用文件夹。其中文件夹的名称指定时不允许有空格,如果 OK 按[Next]键。

(5) GNUPro Toolkit

选择建立 GNU 方法。通常使用默认即可,所以按[Next]键。

(6) GNUPro Toolkit Documentation and Source Code

选择建立文档及源文件。对缺省文件仅选择[HTML 文档]编制,而在

既想参照包括 GCC 的全部源代码,又希望改进的场合,也选择[源代码],按[Next]键。另外,如果没有建立源代码,可以节约 200M 字节左右的磁盘空间。

(7) Source-Navigator

Cygnus 的 C 语言专用引导程序的安装。必需专有的安装键,实际上不可能建立。避开所有的校验,按[Next]键。

(8) Unsupported Tools

不支持开发工具的安装,避开所有的校验,按[Next]键。

(9) Select Program Folder

向 Windows 程序组上登录时设定名称。原封不动地按[Next]键。

按照上述顺序开始安装文件。在安装文件过程中,提示不断打开,直到出现安装结束的对话框,请耐心等待。



• 画面 5.5

安装后在硬盘的根目录上产生[Cygnus]文件夹,是英文,其中 GCC 的文档也可以用 HTML 参照。

3. 试使用 GCC 和 GDB

一旦建立文件之后,在顺序按启动按钮→[程序(P)]集合→[Cygnus]集合中存在的[Cygnus GNUPro Toolkit gnupro-99r1pl]时,GCC 环境设定,指示启动。大部分的工具是指令式的程序,只能是在 Windows 配置下的 MS-DOS 环境下执行,或者只能利用

指令提示。值得注意的是在单个 MS-DOS 环境下不能工作。另外,这个提示终止时,输入[exit]键(参见画面5.6所示)。



• 画面 5.6

• 目录 1 (hello.c)

```
#include <stdio.h>

int main(void)
{
    printf("HELLO H8! \n");
}
```

安装后的工具大部分附带[h8300-hms]的接头词,所以用附带文档记录[gcc]时,必需代换成[h8300-hms-gcc]读出。

那么,我们马上使用 GCC 看看,先制作一些可确认的简单程序,进行编译试试看。这里还不需要 H8 微型计算机插件板。使用记录目录等的文本编辑程序,希望输入目录 1 的程序。这里在 C 盘制作 TEST 文件夹,将程序保存在这个文件夹内,这是进行上

述编译等作业的前提条件。

一旦输入终止,用 hello.c 的文件名保存在[C:\¥TEMP]文件夹内,使用编辑工具做成的文件,保存时[文件种类]选择用默认的[文本文件(*.txt)],所以变更[全部文件]之后,在[文件名]的地方输入[hello.c]按[保存]键。如果不这样作,往往文件名的扩展面会发生变化。

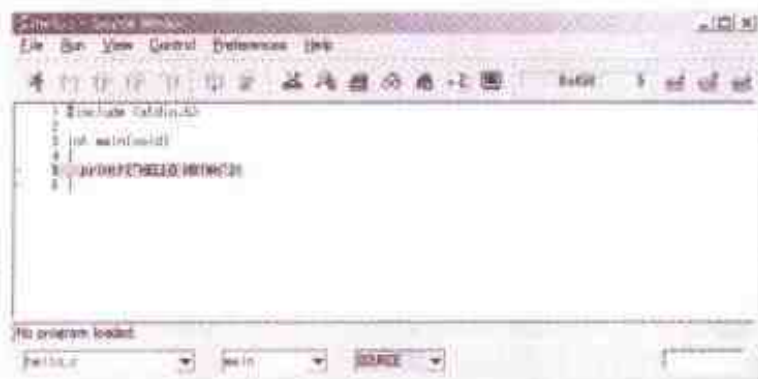
那么,起动 [Cygnus GNUPro Toolkit gnupro-99npl]将程序向保存文件夹(C:\¥TEMP)移动,试编译一下看。

当[h8300-hms-gcc-mh-g hello.c-o hello]输入后,开始编译。如果出现错误,用编辑软件修改原始文件,再尝试一次。当编译成功时,不出现任何信息而终止,且在相同文件夹上如果产生[hello.exe] OK。还有,生成的 EXE 文件在 Windows 上绝对不能运行。最坏的时候,PC 会出现死机的现象(参见画面 5.7 所示)。



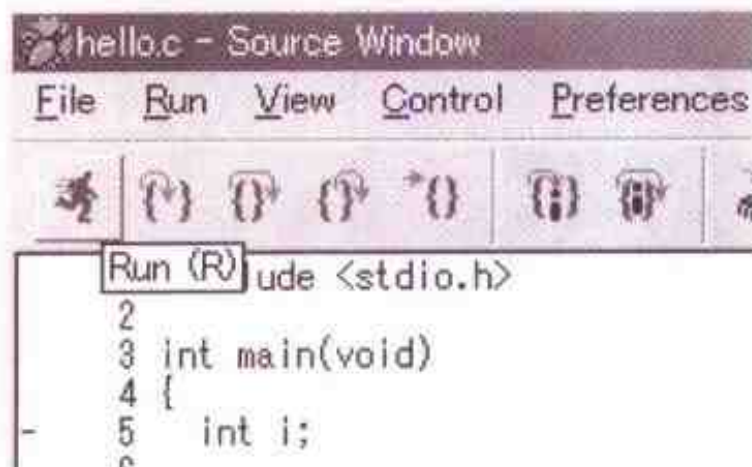
• 画面 5.7

代替 H8 微型计算机插板,在附带 GNUPro Toolkit 的 GDB(原程序编译器)的 H8 仿真程序上运行一下看,这个 GDB 也是 GNU 的一种工具,而只是限定在 Windows 环境的图形方法上它是非常强有力的。刚才由提示希望输入[h8300-hms-gdbhello]。起动输入程序的编译器(参见画面 5.8 所示)。



• 画面 5.8

从这里开始稍微有点麻烦,按原始窗口速度键■,事先起动控制台窗口。这个窗口是进行实际状况的运行记录和打印等仿真控制台输出(本来 GDB 仅提供在这个控制台上的操作及显示功能)(参见画面 5.9 所示)。



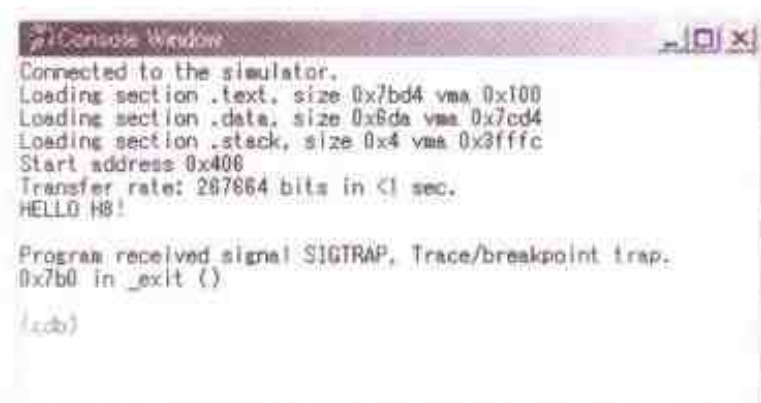
• 画面 5.9

按执行键■时,显示选择目标对话,所以在目标上选择仿真程序(参见画面 5.10 所示)。另外,事先避开右半部分 4 个校验框,然后按 OK 键。



• 画面 5.10

如果显示[HELLOH8]OK,希望以后适当地使用试试看。和源程序一起也可以表示汇编代码,又可以表示寄存器或存储器的监视器。只要这里有,仅进行 C 语言程序设计学习时就足够了(参见画面 5.11 所示)。



• 画面 5.11

4. 在 H8 上运行

这次在计算机上运行试试看,即使对前面的 hello.c,将使用 printf 文字行显示在仿真程序的画面上,然而 AKI-H8 上没有附带那样的显示器。掌握运行状态容易,在端口上连接 LED,我想由程序使其闪烁。因此,在 H8 的 P4 上连

接 8 个 LED 试使其闪烁。

最初,用连接程序在原始文件上必需分段定义(目录 2)。这个文件就是将分配 H8 片内储存信息和编译程序输出代码定义,它具有非常重要的

作用,同时具有通用性。也进行 H8 中断向量和中断函数的分配,所以将和这里定义相同名称的中断函数如果在源代码上宣布,中断向量将自动地生成。

目录 2 (h8rom.x)

```

OUTPUT_FORMAT("coff-h8300")
OUTPUT_ARCH(h8300h)
ENTRY("_start")
MEMORY
{
    /* 0xc04 is a magic entry. We should have the linker just
       skip over it one day... */
    vectors : 0 = 0x0000, 1 = 0xff
    rom : 0 = 0x100, 1 = 0x1ff00
    ram : 0 = 0xfef10, 1 = 0x1000
    /* The stack starts at the top of main ram. */
    topram : 0 = 0xffefc, 1 = 0x4
    /* At the very to of the address space is the 8-bit area. */
    eight : 0 = 0xffff00, 1 = 0x100
}

SECTIONS
{
    .vectors : |
        /* Use something like this to place a specific functions address
           into the vector table. */
        LONG (ABSOLUTE (_start))          /* 0 Reset vector */
        LONG (ABSOLUTE (_start))          /* Reserved */
        LONG (ABSOLUTE (_start))
        LONG (ABSOLUTE (_start))
        LONG (ABSOLUTE (_start))
        LONG (ABSOLUTE (_start))
        LONG (ABSOLUTE (_start))
        LONG (DEFINED(_int_nmi)? ABSOLUTE(_int_nmi):ABSOLUTE(_start))
        LONG (DEFINED(_int_trap0)? ABSOLUTE(_int_trap0):ABSOLUTE(_start))
        LONG (DEFINED(_int_trap1)? ABSOLUTE(_int_trap1):ABSOLUTE(_start))

        LONG (DEFINED(_int_trap2)? ABSOLUTE(_int_trap2):ABSOLUTE(_start))
        LONG (DEFINED(_int_trap3)? ABSOLUTE(_int_trap3):ABSOLUTE(_start))
        LONG (DEFINED(_int_irq0)? ABSOLUTE(_int_irq0):ABSOLUTE(_start))
        LONG (DEFINED(_int_irq1)? ABSOLUTE(_int_irq1):ABSOLUTE(_start))
        LONG (DEFINED(_int_irq2)? ABSOLUTE(_int_irq2):ABSOLUTE(_start))
        LONG (DEFINED(_int_irq3)? ABSOLUTE(_int_irq3):ABSOLUTE(_start))

```



```

LONG (DEFINED(_int_irq4)? ABSOLUTE(_int_irq4):ABSOLUTE(_start))
LONG (DEFINED(_int_irq5)? ABSOLUTE(_int_irq5):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (ABSOLUTE(_start))

LONG (DEFINED(_int_wovi)? ABSOLUTE(_int_wovi):ABSOLUTE(_start))
LONG (DEFINED(_int_omi)? ABSOLUTE(_int_omi):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_imia0)? ABSOLUTE(_int_imia0):ABSOLUTE(_start))
LONG (DEFINED(_int_imib0)? ABSOLUTE(_int_imib0):ABSOLUTE(_start))
LONG (DEFINED(_int_ov0)? ABSOLUTE(_int_ov0):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_imia1)? ABSOLUTE(_int_imia1):ABSOLUTE(_start))
LONG (DEFINED(_int_imib1)? ABSOLUTE(_int_imib1):ABSOLUTE(_start))

LONG (DEFINED(_int_ov1)? ABSOLUTE(_int_ov1):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_imia2)? ABSOLUTE(_int_imia2):ABSOLUTE(_start))
LONG (DEFINED(_int_imib2)? ABSOLUTE(_int_imib2):ABSOLUTE(_start))
LONG (DEFINED(_int_ov2)? ABSOLUTE(_int_ov2):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_imia3)? ABSOLUTE(_int_imia3):ABSOLUTE(_start))
LONG (DEFINED(_int_imib3)? ABSOLUTE(_int_imib3):ABSOLUTE(_start))
LONG (DEFINED(_int_ov3)? ABSOLUTE(_int_ov3):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))

LONG (DEFINED(_int_imia4)? ABSOLUTE(_int_imia4):ABSOLUTE(_start))
LONG (DEFINED(_int_imib4)? ABSOLUTE(_int_imib4):ABSOLUTE(_start))
LONG (DEFINED(_int_ov4)? ABSOLUTE(_int_ov4):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_dend0a)? ABSOLUTE(_int_dend0a):ABSOLUTE(_start))
LONG (DEFINED(_int_dend0b)? ABSOLUTE(_int_dend0b):ABSOLUTE(_start))
LONG (DEFINED(_int_dend1a)? ABSOLUTE(_int_dend1a):ABSOLUTE(_start))
LONG (DEFINED(_int_dend1b)? ABSOLUTE(_int_dend1b):ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (ABSOLUTE(_start))

LONG (ABSOLUTE(_start))
LONG (ABSOLUTE(_start))
LONG (DEFINED(_int_er0)? ABSOLUTE(_int_er0):ABSOLUTE(_start))
LONG (DEFINED(_int_rx0)? ABSOLUTE(_int_rx0):ABSOLUTE(_start))
LONG (DEFINED(_int_tx0)? ABSOLUTE(_int_tx0):ABSOLUTE(_start))
LONG (DEFINED(_int_tel0)? ABSOLUTE(_int_tel0):ABSOLUTE(_start))
LONG (DEFINED(_int_er1)? ABSOLUTE(_int_er1):ABSOLUTE(_start))
LONG (DEFINED(_int_rx1)? ABSOLUTE(_int_rx1):ABSOLUTE(_start))
LONG (DEFINED(_int_tx1)? ABSOLUTE(_int_tx1):ABSOLUTE(_start))
LONG (DEFINED(_int_tel1)? ABSOLUTE(_int_tel1):ABSOLUTE(_start))

LONG (DEFINED(_int_ad)? ABSOLUTE(_int_ad):ABSOLUTE(_start))

FILL(0xff)

```

```

    { > vectors
.text 0x100, {
    * (.text)
    * (.strings)
    * (.rodata)
    _etext = . ;
} > rom
.tors: {
    __ctors = . ;
    * (.ctors)
    __ctors_end = . ;
    __dtors = . ;
    * (.dtors)
    __dtors_end = . ;
} > rom
.data : AT(ADDR(.tors) + SIZEOF(.tors)) {
    _data = . ;
    * (.data)
    * (.tiny)
    _edata = . ;
} > ram
.bss : AT(LOADADDR(.data) + SIZEOF(.data)) {
    _bss_start = . ;
    * (.bss)
    * (.COMMON)
    _end = . ;
} > ram
.stack : {
    _stack = . ;
    * (.stack)
} > topram
.eight : {
    * (.eight)
} > eight
.stab 0 (NOLOAD) : {
    [.stab]
}
.stabstr 0 (NOLOAD) : {
    [.stabstr]
}
}

```

必需启动例行程序(目录 3)。不是特别写出的内容,而是用 H8 汇编程序内容制作的程序,也是通用的程序。

目录 3 (h8cr10.s)
; h8/300 and h8/300h start up file.

h8300h

```

.section .text
.global _start
_start:
    mov.l #_stack, sp
    ; copy DATA from ROM
    mov.l #__dtors_end, er0
    mov.l #_edata, er1
    mov.l #__data, er2
loop0: mov.w @er0, r3

```



```

mov.w r3,@er2
adds #2,er0
adds #2,er2
cmp.l er1,er2
blo .loop0

;clear BSS
mov.l #_edata,er0
mov.l #_end,er1
mov.w #0,r2 ; not
sure about alignment requirements
loop: mov.w r2,@er0
playing it safe for now
adds #2,er0
cmp.l er1,er0
blo .loop
jsr @__main

```

```

jsr @__main
jsr @__exit

.section .stack
_stack: long 1

```

那么,本题是 C 语言程序(目录 4)。H8 内部 I/O 定义的标题文件(3048F.H)附带在秋月电子出售的 AKI-H8 用 C 编译程序上,可以原封不动地利用,而这里最低限度要求在 P4 端口源程序内进行定义,它可以由日立网页下载。

目录 4(led1.c)

```

struct st_p4 { // P4 结构体定义
    unsigned char    DDR;
    char            WK1;
    union {
        unsigned char    BYTE;
        struct {
            unsigned char    B7 :1;
            unsigned char    B6 :1;
            unsigned char    B5 :1;
            unsigned char    B4 :1;
            unsigned char    B3 :1;
            unsigned char    B2 :1;
            unsigned char    B1 :1;
            unsigned char    B0 :1;
        }
    }
    char            wk2[18];
    union {
        unsigned char    BYTE;
        struct {
            unsigned char    B7 :1;
            unsigned char    B6 :1;
            unsigned char    B5 :1;
            unsigned char    B4 :1;
            unsigned char    B3 :1;
            unsigned char    B2 :1;
            unsigned char    B1 :1;
            unsigned char    B0 :1;
        }
    }
}

```



```

    }
    BIT;
    PCR;
};

#define P4 ( *(volatile struct st_p4 *) 0xFFFF05) // P4 的宏指令定义

int main(void)
{
    int i;
    char d;

    P4.DDR = 0xff; // // 全位输出设定

    while (1) { // 无限循环
        for (d = 0; d < 255; d++) {
            P4.DR.BYTE = d; // 在 P4 上写入数据
            for (i = 0; i < 30000; i++) ; // 循环浪费(权数)
        }
    }
}

```

输入目录 2~4, 如果保存在[C:\
¥TEMP]文件夹上, 则启动 GNUPro
Toolkit 控制台输出, 移动至 [C:
¥TEMP]文件夹中。

输入的命令是:

C:\¥WINDOWS>CD¥TEST

下面是运行编译程序。

C:\¥TEST>h8300-hms-gcc -mh
-mrelax -O0 -nostartfiles h8rom. x
h8crt0.s led1.c -o led1

这里需要注意的是, 给予编译程
序选择或原始文件的名字不要将大、
小写字母混同。由于编译程序区分
大、小写字母。另外, 空格也通过选择
或是需要, 或是不需要, 将各选择大概

的意思表示如下:

-mh	输出 H8/300H 用的代 码
-O0	(O-0), 由于过分最优 化使程序变得没有意 义, 所以抑制最优化 程度。
-nostartfiles	连接时, 没有使用标 准系统起动文件。
-o<...>	将(O)<...>作为最终 输出文件名。

我想读者已经察觉, 在编译时, 从
连接程序用文件到汇编程序文件整个
一起记述。GCC 就是通过判断文件的
扩展名, 自动地起动汇编程序或连接
程序。

编译成功时, 完成[led1.exe]文

件。即使写入 H8,这也不是可执行的文件形式。用所谓 objcopy 工具可进行文件转换。即:

```
C: ¥TEST > h8300-hms-objcopy-O
srec -R.stack led1.exe led1.mot
```

这里完成摩托罗拉 S 格式(hello.mot)文件。以后仅仅写入这个文件,然而,由于对包括秋月电子的 AKI-H8 开发配套元件[FLASH.EXE]没能写入(日立单独的格式),所以使用带有 CD 的 FW.EXE(P. 56),传送给 H8。另外,希望二进制图形的情况,如果指定由二进制文件完成。这些也用 FW.EXE 一一对应写入。

```
C: ¥TEST > h8300-hms-obj-
copy-O binary -R.stack led1.exe
```

led1.bin

正常情况,编译、转换如果能写入 H8,我们马上运行试试看。在 LED 上数字递增的状态一定逐一表示出来。

这次使用的 GCC,在市场上出售的编译程序在价格方面对个人用户是相当高的,对学校等将编译程序复制在各个 PC 微型计算机上使用时,如果特许方面等有问题时,请采用免费并且强有力的编译程序。GCC 有在多种 OS 环境下可以利用的编译程序,不限于 Windows,往往也可以将开发环境移置到 Linux 等上管理。

以后使用 C 语言的实例等以使用 GNUPro Toolkit 为前提进行,所以希望读者认真阅读本章,并希望事先抓住感觉。

chapter

6

机器人的嘴

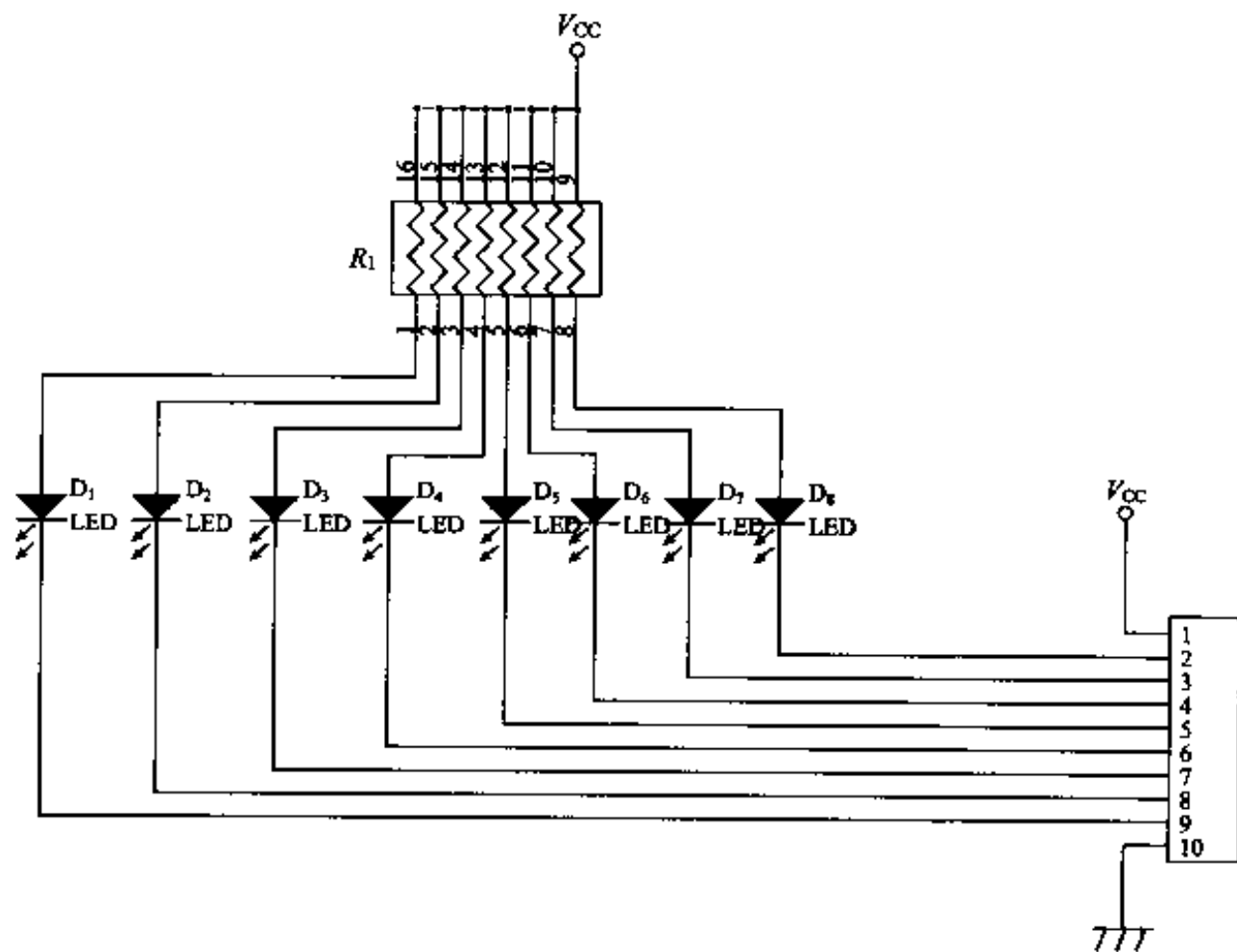
为了表示机器人想要说什么，只要安装发光二极管（LED）或显示器，即能发出单纯的响声。由于在H8上有DA转换器，所以用它也可以发出有趣的声音。

这样，如果有来自机器人的信息，就可以了解机器人的程序是否在正确工作，或者机器人行走的状态变得如何等等。

6.1 在 LED 上表示

在发光二极管(LED)上输出 8 位数据时,就能表示二进制的数据。由此可以判断微型计算机是否在正确地工作。当然可以将微型计算机想要说

的话表示在 LED 上,图 6.1 所示的就是连接在微型计算机上的 8 个 LED 的电路。



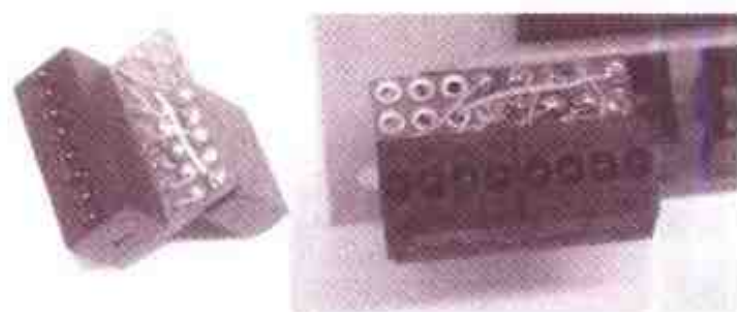
• 图 6.1 8 个 LED 的电路

图 6.1 中 R_1 是电阻阵列,如照片 6.1 所示,是在一个部件上装上 8 个电阻。由于微型计算机用 8 位处理数据,所以 8 个电阻可以看成更多的处理数据。

另外,也有 LED 和电阻部分成为一体的部件。照片 6.2 就是将它们连接在连接器上成为一体的情况。通过将它们连接在微型计算机的出口上,可以检验输出口或在 LED 上显示。



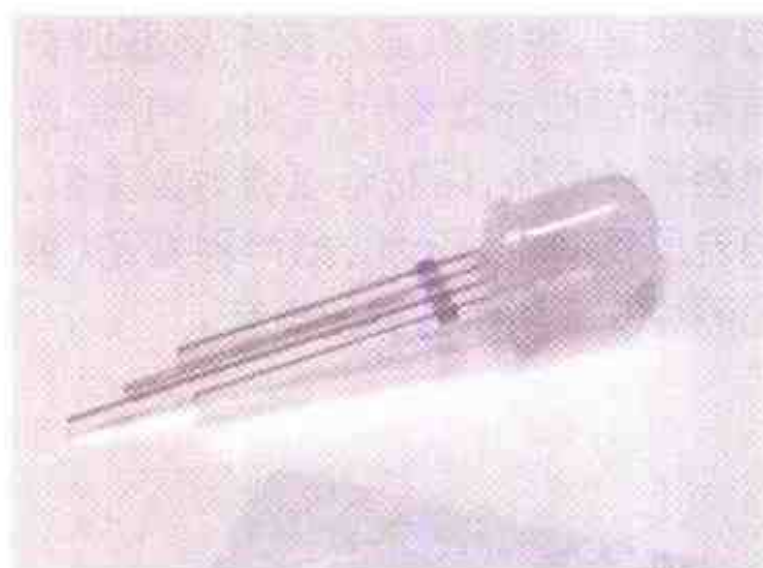
• 照片 6.1 电阻阵列



• 照片 6.2 LED 和电阻部分成为一体部件

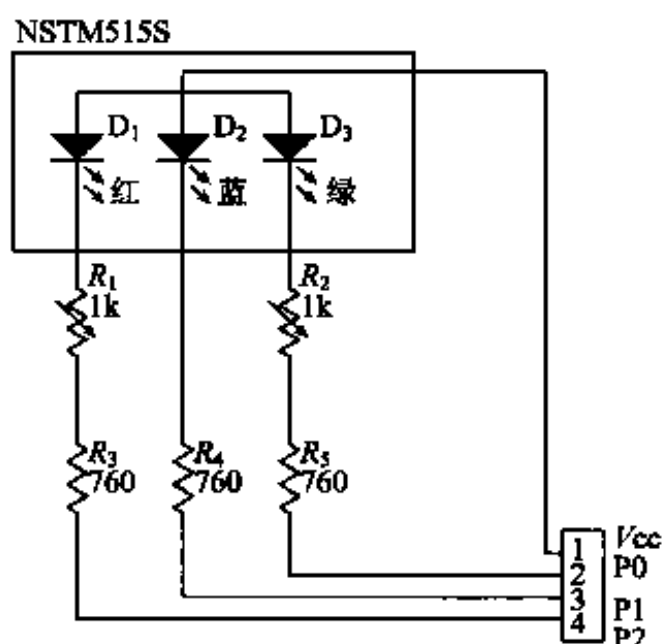
6.2 全色 LED

照片 6.3 是全色发光元件。事先装好红、蓝、绿色的 LED, 通过改变光亮度, 使其发出各种各样的颜色。对于大画面荧光屏等, 这种二极管可以完全充满整个荧屏。

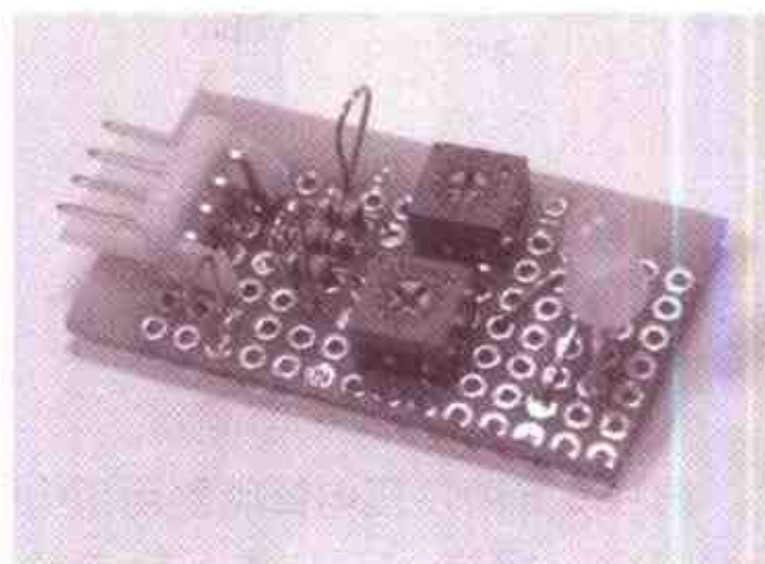


• 照片 6.3 NSTM5158

使用方法如图 6.2 所示, 与使 LED 发光相同, 然而根据 LED 的特性, 因各自的颜色不同亮度变得不同, 所以用可变电阻调整各个电流值, 调整好平衡, 使全部为 ON 状态时为白色。



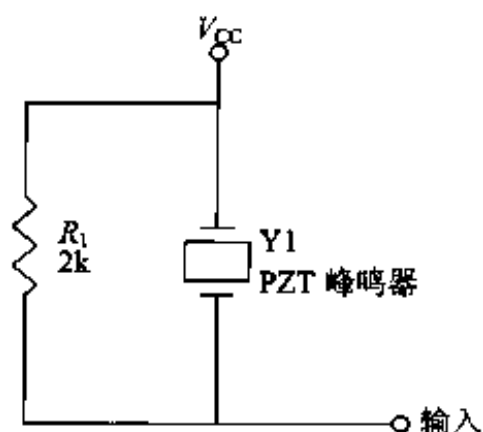
• 图 6.2 全色 LED



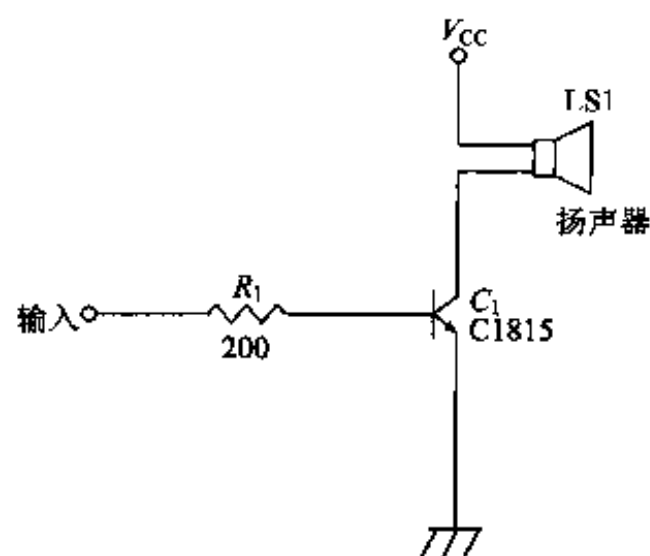
• 照片 6.4 完成的插件板

6.3 让压电蜂鸣器或扬声器发出声音

为了用蜂鸣器或扬声器发出声音,用图 6.3、图 6.4 所示电路,只要将其连接在微型计算机的输出端口上就可以。压电蜂鸣器很简单,可以从端口直接驱动。扬声器需要用晶体管驱动,它们都是通过方波发出声音,很难变更音色。



• 图 6.3 压电蜂鸣器的驱动电路



• 图 6.4 扬声器的驱动电路

照片 6.5 所示压电蜂鸣器用压电元件(PZT)构成。PZT 是一种电容器,声音高时,PZT 电荷向电阻放电,

声音低时向 PZT 充电。由此,PZT 压电扬声器反复充电、放电形成振动,这样就发出声音。

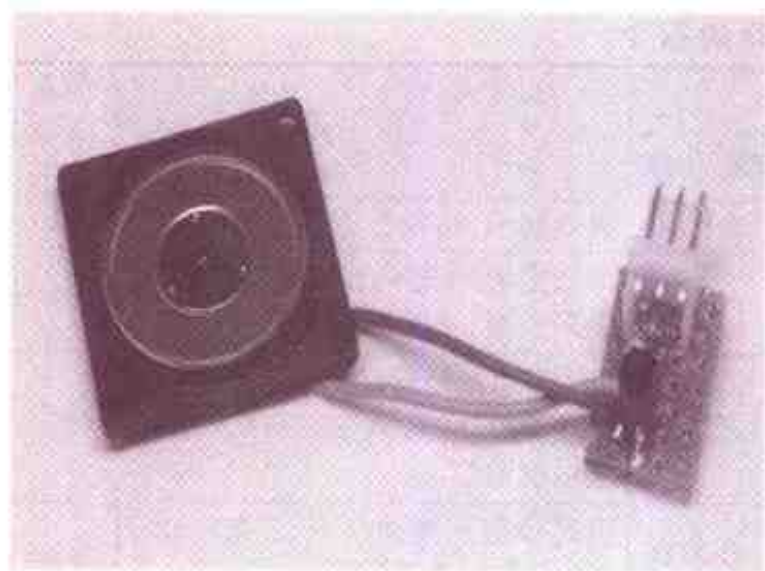


• 照片 6.5 压电蜂鸣器

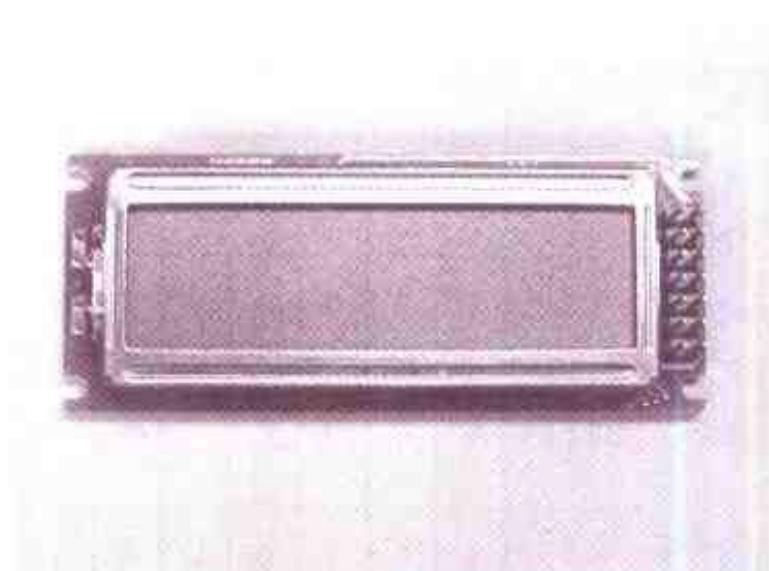
扬声器(照片 6.6)就是在磁场中放置线圈,当电流流过这个线圈时根据弗莱明左手定则产生力。由此,扬声器振子位移,切割电流时返回原位,电流反复 ON-OFF 时,扬声器振动(参见照片 6.7 所示),发出声音。



• 照片 6.6 扬声器



• 照片 6.7 扬声器和驱动回路



• 照片 6.8 LCD 显示器

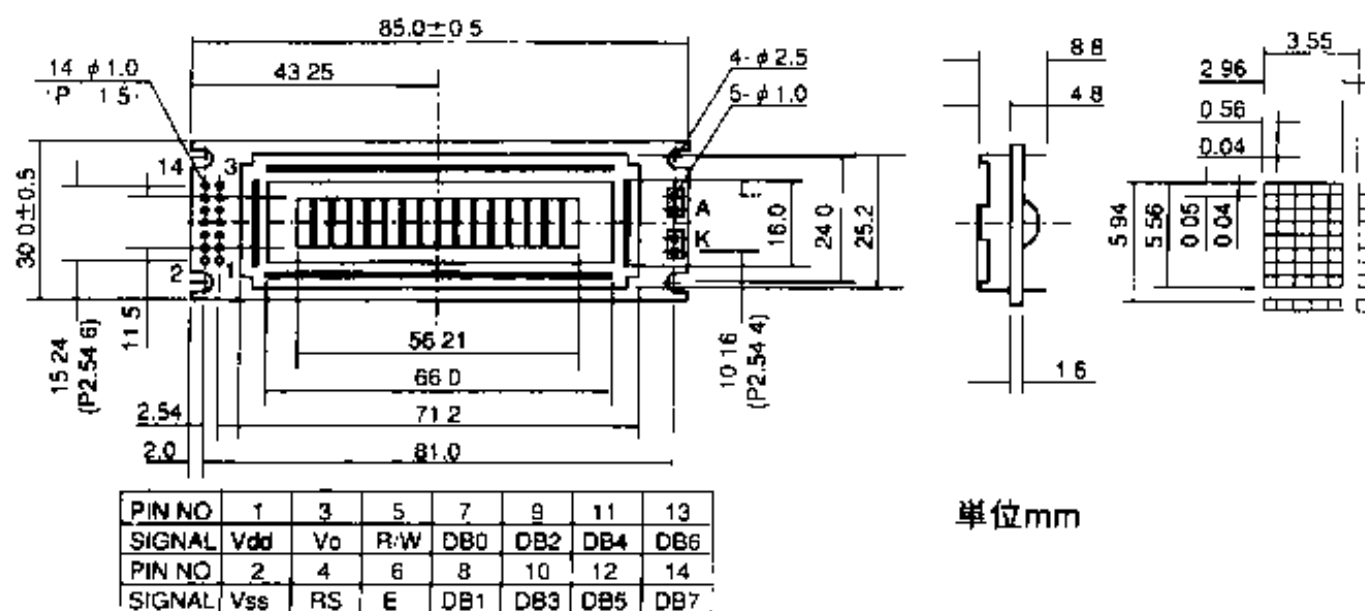
6.4 液晶显示器(LCD)

在液晶显示器(LCD)上可以表示文字或数字。对于 I/O 插板上的 LED, 显示是二进制的, 人们理解需要时间, 而 LCD 是用字符显示, 所以容易理解。另外, 对串行通信, 为使机器人动作, 通信电缆会成为障碍, 在实际行走时可以不用它。为此, 可以说 LCD 是面向实战的显示装置。在程序开发的初期, 可以使用串行通信 (RS232C), 完成度提高后再使用 LCD

等希望提高效率的方法。液晶显示器是 16 文字 × 2 行的 LCD 字符显示器 (SC1602BSSLB)。

LCD 外形尺寸和连接接插件引脚配置如图 6.5 所示。

连接器有电源、8 位数据总线、R/W 信号等, 有像存储器一样使用的 LCD 模块。液晶显示器, 根据表 6.1 所示命令可以进行清零及复位等操作。



• 图 6.5 LCD 外形尺寸及连接接插件

表 6.1 LCD 的指令

指 令	代 码										说 明	最大 处理 时间
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	清 0 显示,将指针返回原来位置。	1.64ms
Cursor at home	0	0	0	0	0	0	0	0	1	.	不变更显示的内容,将指针返回原来位置,*不影响它的位。	1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	I/D 表示指针移动方向。I/D=0,增加,I/D=1,减少。S=1 进行显示器进位。	40 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	设定 D 显示器 on/off,C 指针 on/off、B 指针有无闪烁。	40 μ s
Cursor/Display Shift	0	0	0	0	0	1	S/CR/L	.	.	.	S/C=1,显示器进位、S/C=0,指针偏移。R/L=1 向右偏移、R/L=0 向左偏移。	40 μ s
Function Set	0	0	0	0	1	DL	N	F	.	.	D/L=1 变为 8 位数据端口,0 变为 4 位数据端口。N=1,2 行显示,N=0,1 行变为显示。F=1 表示 10 位文字,F=0 表示 7 位文字。	40 μ s

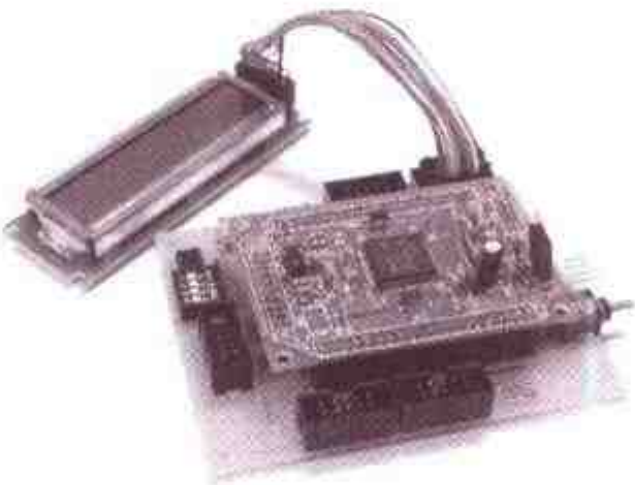
6.4.1 和 H8 的连接方法

LCD 连接在 H8-3048F 的端口 4 上使用。在 H8-3048F 端口 4 的 I/O 插件板的接插件上输出,所以在这里决定连接使用 LCD。与 LCD 的连接如表 6.2 所示。其中,数据总线仅连接 4 位,通过将上位、下位的 4 位分 2 次传送,将一个字节的数据表示在 LCD 上。

照片 6.9 示出 I/O 插件板和 LCD 的连接(在 LCD 上安装插孔)。在配线时,特别要注意电源的连接,在插入插孔之前,要再一次用万用表等确认是否连接好。对用于实战用机器人上的,需用丙烯酸板等保护好。

表 6.2 I/O 插件板和 LCD 的连接

I/O 插件板 引脚号	CPU 信号	LCD 引脚号	LCD 信号
1	+5V	1	Vdd
2	P4-7 位	—	—
3	P4-6 位	—	—
4	P4-5 位	4	RS
5	P4-4 位	6	E
6	P4-3 位	14	DB7
7	P4-2 位	13	DB6
8	P4-1 位	12	DB5
9	P4-0 位	11	DB4
10	GND	2,3,5	Vss,V0,R/W



• 照片 6.9 LCD 和 I/O 插件板的连接

6.4.2 程 序

在 LCD 显示器上显示文字和 16 进制数的程序,显示 Hello! I am a Robot 以后,将数字用 16 进制数递增记数。

按照操作说明书进行程序设计时,实例程序 LCD 变为初始化程序。

另外,4 位数据输出例行程序也相同。这是依赖硬件的部分,所以包括等待程序最好不要变更。对于这个程序,由于 LCD 处理滞后,所以多用等待程序。为了显示需要一定时间,所以仅限于在机器人程序时间有剩余时使用。LCD 读取存入只使用 CPU 时间。

6.5 由扬声器发出声音

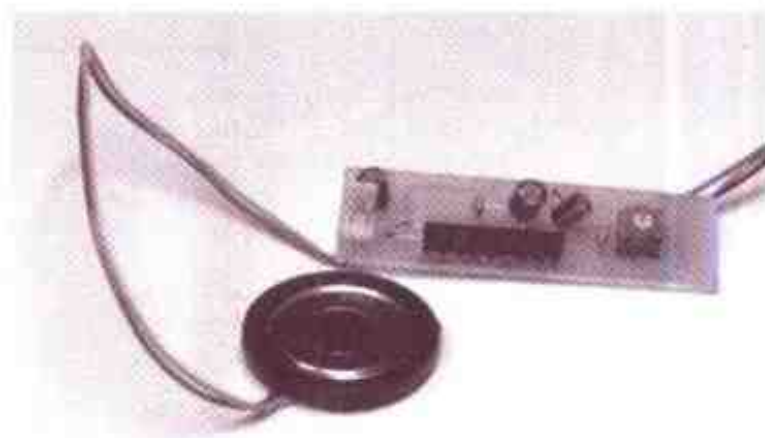
对 H8 的 8 位微型计算机,DA 转换器可以使用 2 通道,从其中的 1 通道试看输出正弦波形。

为了使扬声器发出声音,必需放大它的输出。使用一个芯片的放大器(双标低压功率放大器)NJM2073(参见照片 6.10 所示)。

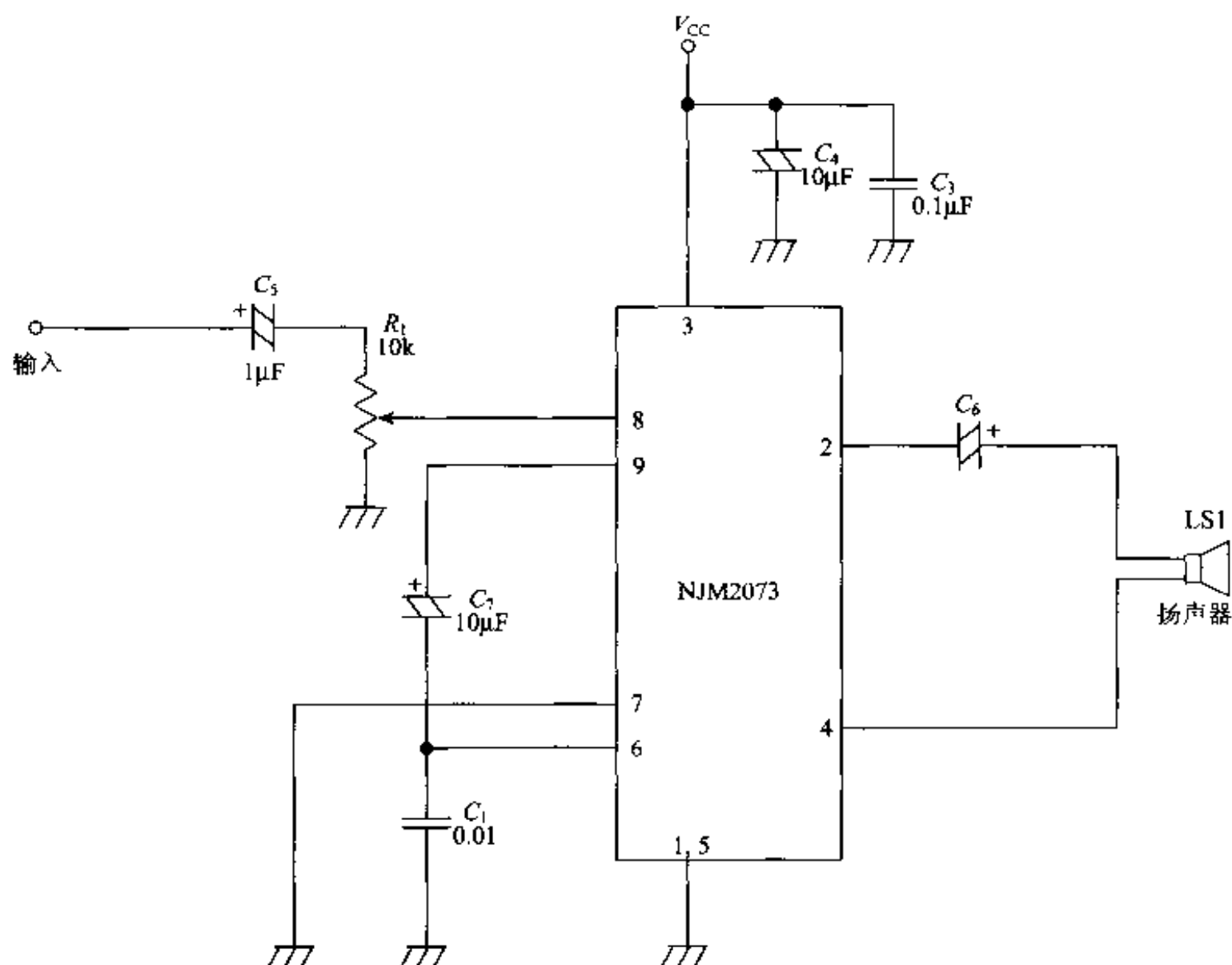
电路图如图 6.6 所示,在 input 部分上连接 DA 转换器的输出。

NJM2073 是 2 回路输入,低电压工作的功率放大器,它可以从下述新日本无线网页下载数据图表:

http://www.njr.co.jp/_jfr005sem.htm



• 照片 6.10 扬声器放大器



• 图 6.6 扬声器放大器电路

6.6 在个人电脑上通过通信端口输出文字和数字

使用 RS232C, 可以把微型计算机将数据或文字表示在个人电脑上时, 程序的调试是容易的。另外, 可以判断机器人以什么样的状态工作。

用 RS232C 电缆连接微型计算机和个人电脑, 试执行以下程序。有关微型计算机侧和个人电脑侧的程序讲解如下。

1. 微型计算机的程序

实例程序就是从 H8-3048F 向个人电脑上输出文字的程序。有关 SCI

在第 3 章中已经叙述, 而这里是进行其程序设计、串行端口初始设定和文字输出部分。理解好如何进行通信时, 在使用个人电脑上的调制解调器等连接因特网时也起作用。

下面介绍将文字或数字简单地变换文字行数据的 sprintf 函数程序。

和 C 语言的 printf 函数的使用方法相同, 可以记述文字或数字, 所以很方便。

如果参照程序的主例行程序, 我

想如果有使用过 printf 函数的人们对这种方法会容易理解。

2. 微型计算机和个人电脑的通信方法

下面叙述用串行通信连接微型计算机和个人电脑的方法。

(1)将上述的程序编译,并且在 H8-3048F 上写好,然后连接通信电缆。由于通信电缆在写入 H8 的快速存储器(Flash ROM)时已经使用了,所以这种状态即可。

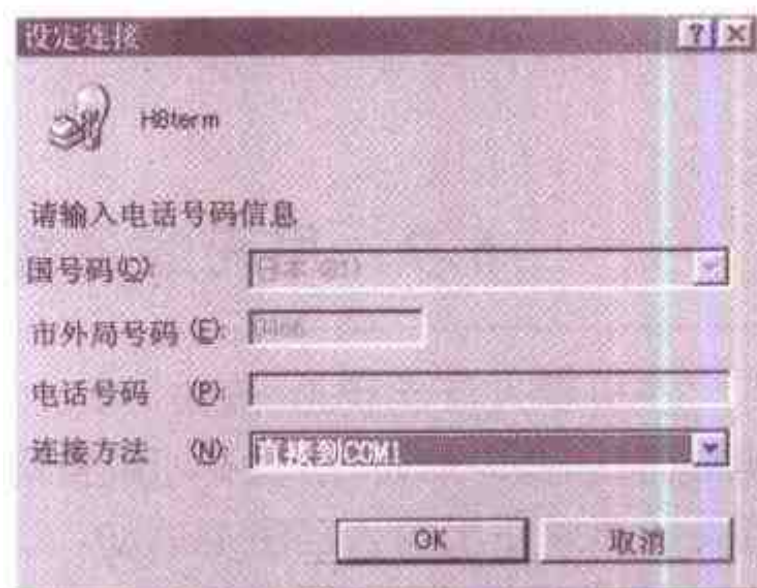
(2)进行个人电脑的设定,使用带有标准的 Windows95/98/2000 的多个终端。从 Windows 画面起动→程序→辅助设备→通信→多个终端的顺序执行,当执行 hypترم.exe 时形成画面 6.1。同时通过输入适当的名字,也可以选择目标。



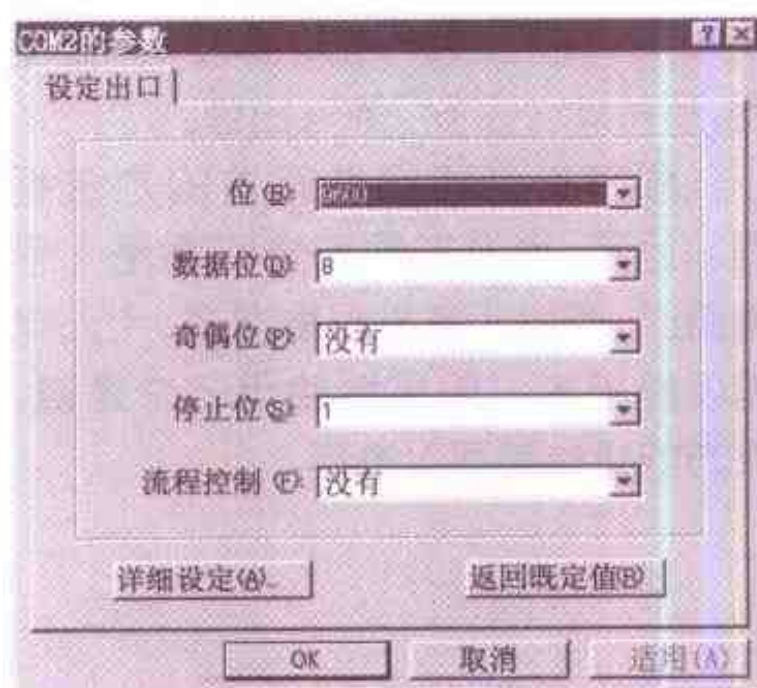
• 画面 6.1 设定连接 1

(3)设定连接方法。选定自己计算机通信端口(RS232C),通过 COM1、COM2 和计算机自身的很多个端口,而这个通信端口必须和 H8-3048F 的通讯端口连接。只要选定在 H8-3048F 的程序写入上使用的端口就可以。

(4)设定 COM 端口的参数。其中,如在上节说明的设定串行通信的通信速度、数据位、奇偶位和停止位。画面 6.3 的数值和 H8-3048F 中设定的实例程序 1 的数值相同。这里个人电脑和机器人通信条件应该一致。



• 画面 6.2 设定连接 2



• 画面 6.3 COM 出口参数的设定画面

将上述设定结束时,下面变成很多个终端画面。确认机器人和个人电脑之间的通信电缆正确连接后,插入

H8-3048F(I/O)的电源。

如画面 6.4 所示,如果在很多个终端画面上出现 Hello! 的文字,程序和个人电脑设定为 OK。不显示时,不仅校验程序,而且对个人电脑设定或通信电缆等也要进行校验。

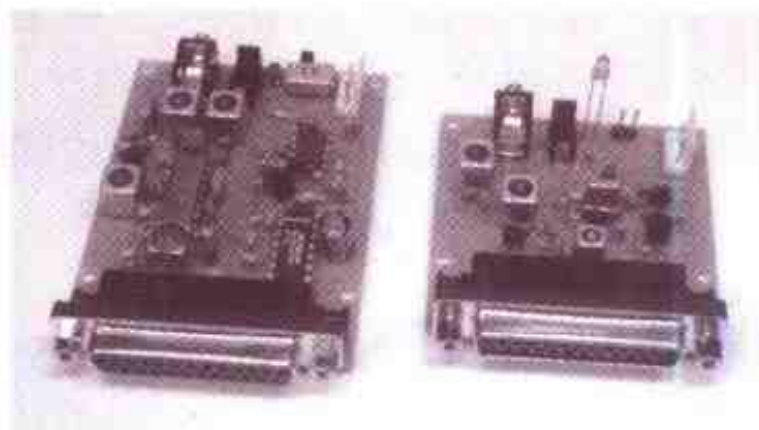


• 画面 6.4 很多个人终端画面

6.7 用无线遥控器驱动机器人奔跑

在机器人行走状态,连接 RS232C 电缆时,已变为障碍,所以不可能按照原来的想像使机器人自由地运动。这时,好像用无线遥控器通过修改数据可以驱动机器人奔跑。无线发送、接收器使用市场上出售的即可。照片 6.11 是 SANDEN 商社装配的无线发送、接收器,还有其它许多厂家在出售。

顺序和上述归纳叙述的方法相同。但是,当发送器和接收器是 1 个通道时,微型计算机仅能发送,个人电脑仅能接收。如果能使用 2 个通道,双方的通信都可以进行。



• 照片 6.11 无线发送、接收器

经销商信息

SANDEN 商社

FAX: 03-3251-1858

chapter

7

机器人的五感

即使是机器人，也和人一样需要有感觉。为了能够适应周围状况的变化，机器人的这种感觉越多越好。

个人电脑是通过键盘或鼠标器给出指示，机器人也是这样的装置，在它上面安装各种传感器，根据从那里获得的信息，可以使它作出各种各样的动作，信息越多越好。

7.1 按钮开关

通常,按钮开关在使机器人开始动作时使用。当按下按钮时,为了起动机器人,只要执行程序即可。另外,停止机器人时也可以使用。按钮在其它方面用途很多,这里不再重述(参见照片 7.1 所示)。

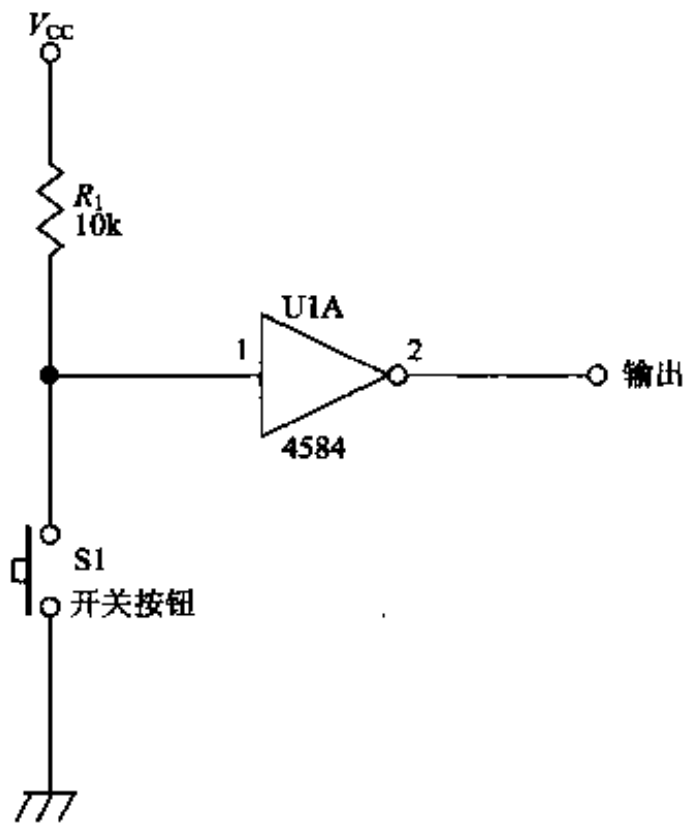


• 照片 7.1 按钮开关

图 7.1 表示为将开关拉起使用 $10\text{k}\Omega$ 左右电阻的电路图,按下也可以使用这个电阻。将这个连接部分,如果连接在 CPU 的输入口上就可以使用。如果连接带有施密特触发器的逆变器,多少可以防止振动,不过最好不要对效果抱有太大的希望。因而,这个逆变器没有也可以。对图 7.1 按下按钮时,4584 的输入端变为 0V ,所以输出变为 5V ,对计算机输入为 1,对于 OFF 状态,输出为 0V 。

1. 程序制作方面的要点

当程序作成时,注意开关的振动。开关输入之后,稍微待一会儿,如果执行程序,好像再一次校验开关一样,就可以防止振动。



• 图 7.1 按钮的输入电路

2. 使用部件的信息

• CMOSIC 4584B 的规格

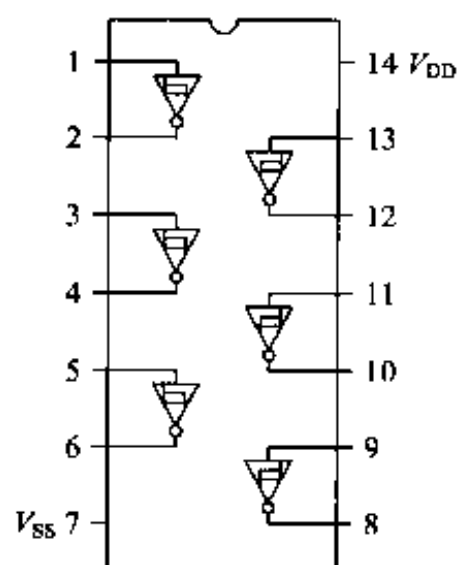
4584 为可以使用电源电压 $3\sim 15\text{V}$ 的带有滞后作用的 NOT 电路。滞后作用为输入、输出电压不同,当输入电压上升、输出电压变低;输入电压下降、输出电压变高。由于噪声,为了使输出不反复高、低变化,也可以使用它防止振动(参见照片 7.2 和图 7.2 所示)。

名词解释

- 拉起:在 5V 电源上应该通过电阻连接。
- 按下:应该通过电阻接地(GND)。
- 振动:开关等机械的触点接触时要反复地进行 ON-OFF 的转换。



• 照片 7.2 4584 的外形



• 图 7.2 引脚连接图

7.2 数字开关

数字开关就是带有很多个的开关元件,通常输入 CPU,8 个开关形成一组的元件经常使用。

如照片 7.3、照片 7.4 所示,有开关型和旋转型,所以选择使用方便开关即可。数字开关对相扑机器人在改变开场模式时使用。另外,对直线追踪机器人,在改变速度或行走方式时使用。

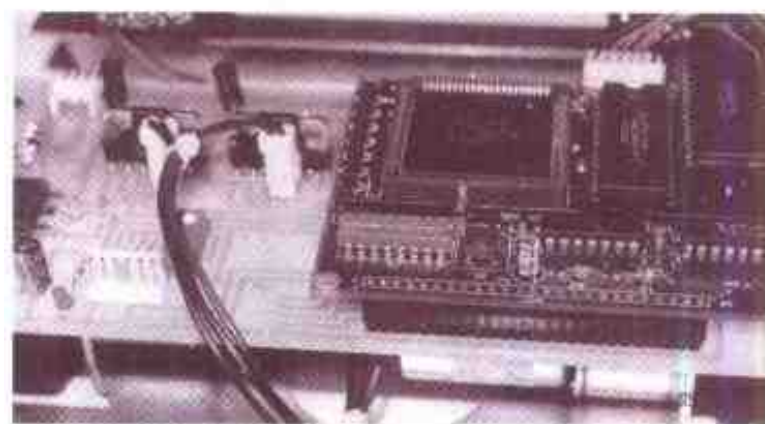


• 照片 7.3 开关型开关

如图 7.3 的连接电路, S1 是数字开关。如照片 7.5 也有将数字开关装配在微型计算机插件板上的情况。



• 照片 7.4 旋转型开关

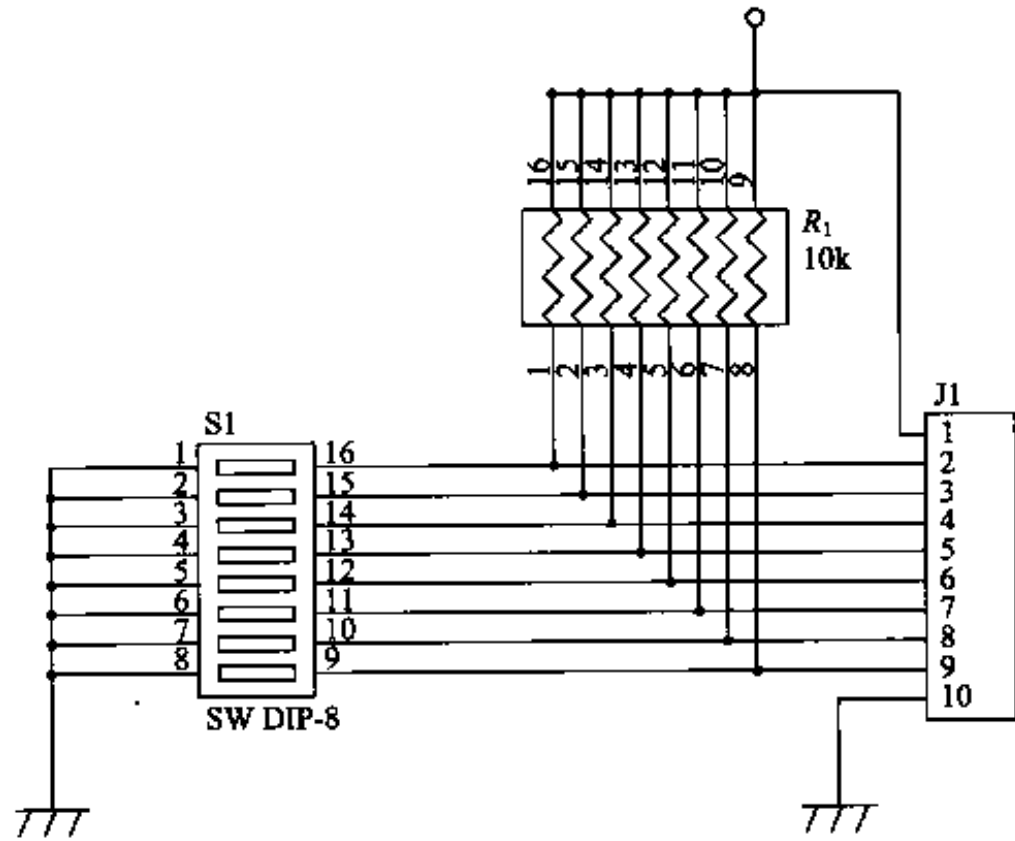


• 照片 7.5 装配在 CPU 插件板的数字开关

程序

将图 7.3 的开关数据输入 H8 的端口程序。由数字开关输入 ON 时为

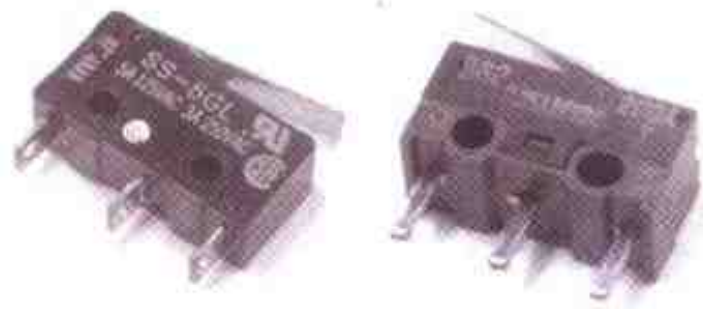
0,这与人们的感觉不符。因此,希望制作将位反转并进行数据处理的函数。



• 图 7.3 数字开关电路

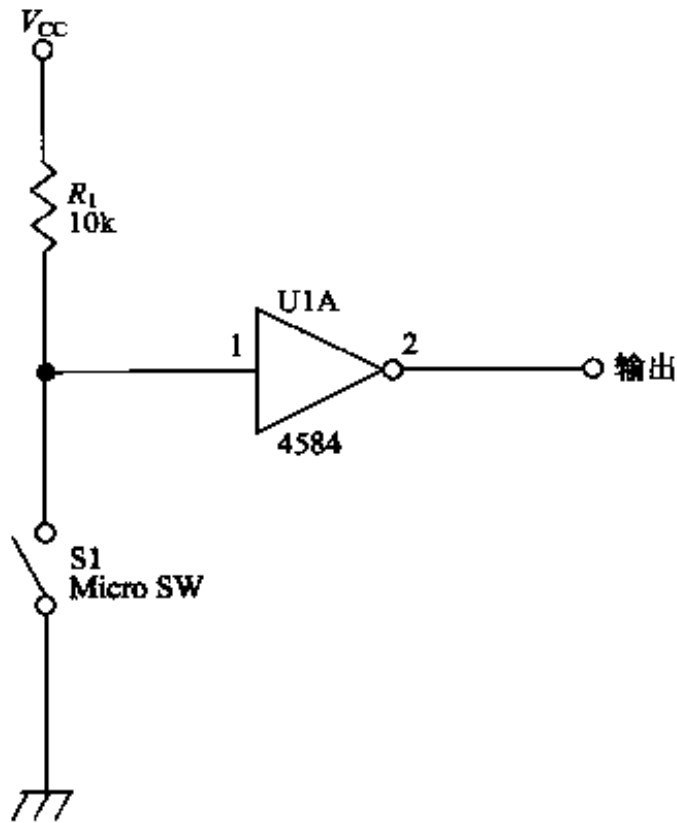
7.3 接触传感器

为了检测接触的物体,使用接触传感器。对相扑机器人使用它检测对手。作为接触传感器,可以使用如照片 7.6 那样的微型开关。



• 照片 7.6 微型开关

在机器人上使用这种微型开关时,产生振动,运用软件使其避开即可。电路图如图 7.4 和按钮开关相同。



• 图 7.4 接触传感器电路

另外,程序也可以和按钮开关相

同。对相扑机器人应该经常校验这个传感器,所以在加速等时开关有任意操作 ON-OFF 的情况。由此,由于机

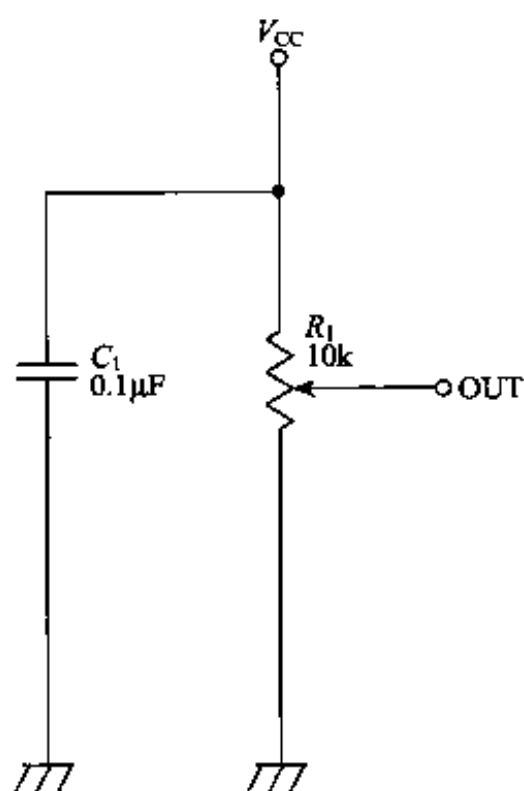
器人有误动作的情况,所以对机器人加速等时,只要停机检测这个误动作产生原因即可。

7.4 由电位器模拟输入

电位器称为可变电阻,就是通过旋钮可以使电阻值变化。将电位器如图 7.5 连接时,可以使输出电压变化。对机器人,使用它(电位器)进行机器人臂或转向位置等检测,它是价格低的角度传感器(参见照片 7.7 所示)。



• 照片 7.7 电位器及其端子



• 图 7.5 电位器的连接电路

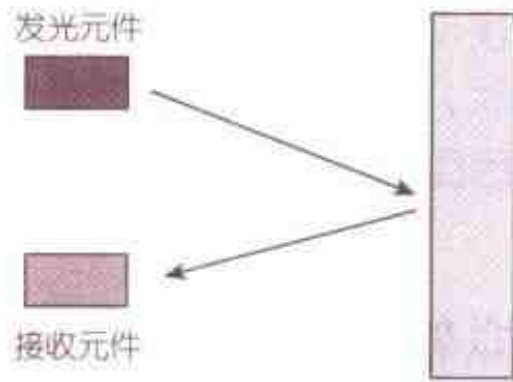
7.5 光传感器

光传感器对相扑机器人,在检测对手或检测摔跤场时使用。对直线追踪机器人,为了检测直线是不可缺少的传感器。一般使用红外线可避开外部干扰光线。还有,对干扰由于变得强烈,所以也使用带有调制回路取消光传感器无光照电流变化那样的方法。



7.5.1 光传感器的原理

对机器人控制使用红外线传感器,如图 7.6 由发光元件发出红外线,用接收元件通过检测其反射光,判断是否有障碍物。对发光元件使用红外线 LED,作为接收光元件使用光电二极管或光电晶体管。



• 图 7.6 光传感器的原理

另外,也有使用可视光的 LED(参见照片 7.8 所示),可视光用肉眼可以确认发光的状态。观察接收光元件感度特性,最好是不要使用频率大幅度偏移的元件。

另外,可视光二极管的定向角也不同,所以必须予以注意将光轴和其一致。如表 7.1 示出不同的辉度和波长。对将红外光电晶体管或红外光电二极管作为接收光元件使用时,使用

近似于红色的 LED EL333-URC-2 红色即可。



• 照片 7.8 可视光的 LED EL333-URC-2 红色

LED 经销商

MARUKA 电机工业株式会社

TEL:03-3255-7509(日本东京)

表 7.1 可视光二极管的种类

型 号	定向角/deg	辉度/mcd	波长/nm	正电压/V	正电流/mA
NSPW-500BS 白色	20	5000	—	3.6	20
E1L51-3b 蓝色	15	1800	465	3.6	20
E1L51-3G 绿色	15	5600	520	3.6	20
EL333-URC-2 红色	25	2000	660	1.7	20
333-USC-C 橘黄色	25	2000	620	2.1	20
EFY-5563 黄色	30	2000	570	2.1	20

• EL333-URC-2 为 Everlight 社的产品

<http://www.everlight.com/main-E.htm>

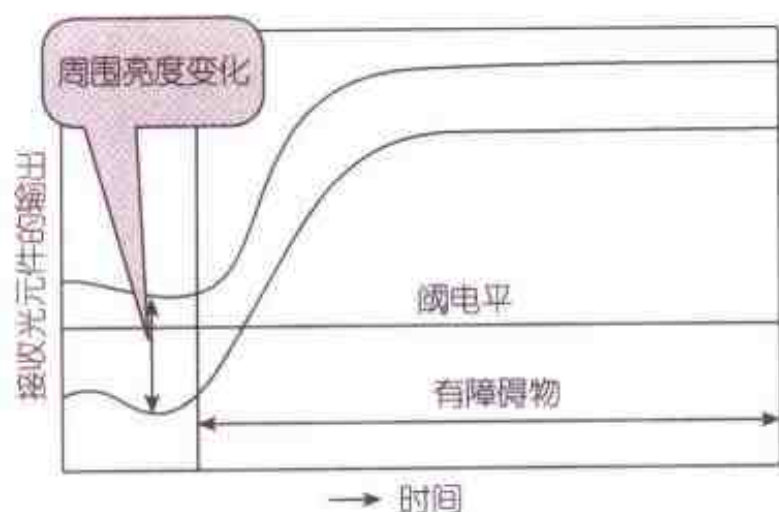
7.5.2 简单的光传感器电路

图 7.8 是简单的反射型光传感器的一个实例,对相扑机器人使用光传感器检测摔跤场。另外对直线追踪机

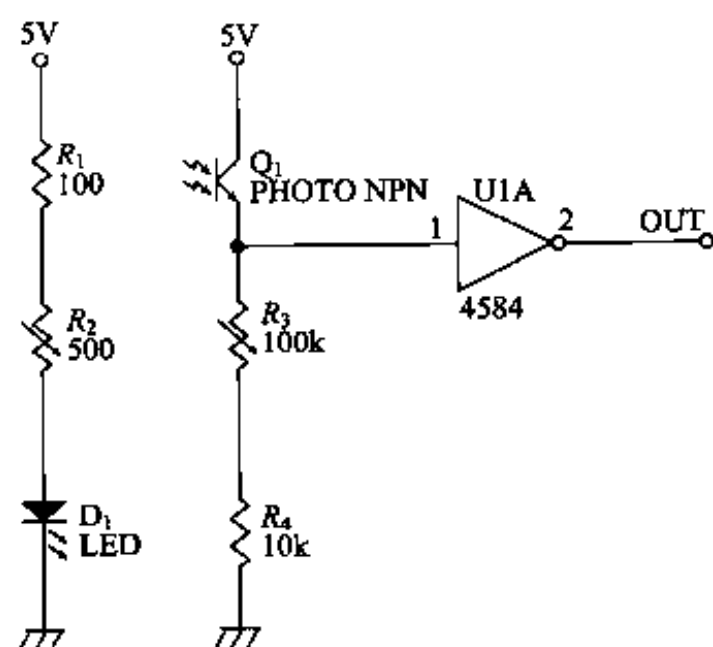
器人,使用光传感器检测白线。这个电路的检测距离对白色的测量物大体是 5~10mm 左右。

这个电路可以容易地制作,然而它对抗干扰光的能力弱。由于发光的反射光以外的光进入传感器,所以输出电

平一定变化。由于这个原因有误动作的情况(参见图 7.7 所示)。4584 的输入变成阈电平以上时,输出变成 OFF。例如,周围亮度变化时,或由于使用红外线的自动聚焦的照相机或摄像机受到影响时,有进行错误输出的情况。



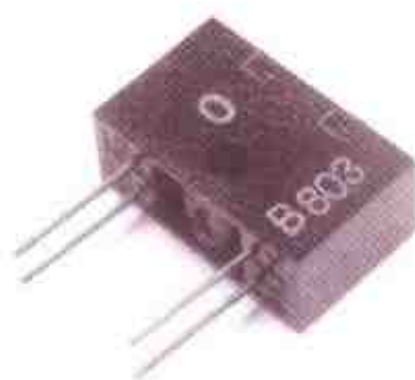
• 图 7.7 由于干扰输出的变化



• 图 7.8 光传感器电路

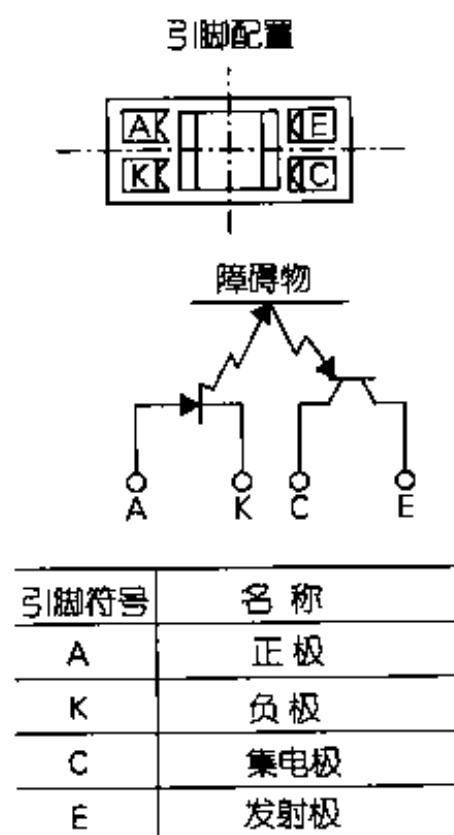
对图 7.8 的电路,D1 使用 LED 的红色光或红外线光的光电二极管。在光电晶体管 Q1 中,使用红外光电晶体管。这时光的波长如果不适合两者,很难获得更大的感度。

如照片 7.9,LED 和光电晶体管作一体化的元件,使用它时,一定符合两者的特性。



• 照片 7.9 欧姆龙的 EE-SFS

照片 7.9 是欧姆龙的 EE-SF5。如图 7.9 的形状,在内部装有 LED 和光电晶体管,引脚配置如图 7.9 所示。



• 图 7.9 欧姆龙 EE-SF5 的说明书

欧姆龙网页

<http://www.omron.co.jp/>

在图 7.8 的电路中,为了使其抗干扰能力更强,用可变电阻调整,尽可能将红外光电晶体管侧的电阻 R_2 值降低,在 LED 侧使电阻 R_3 变小方向上调整即可。由于增大发光量,所以抗干扰光的能力变强。另外,安装外壳也是减少干扰光影响的一种方法。

7.5.3 调制型光传感器电路

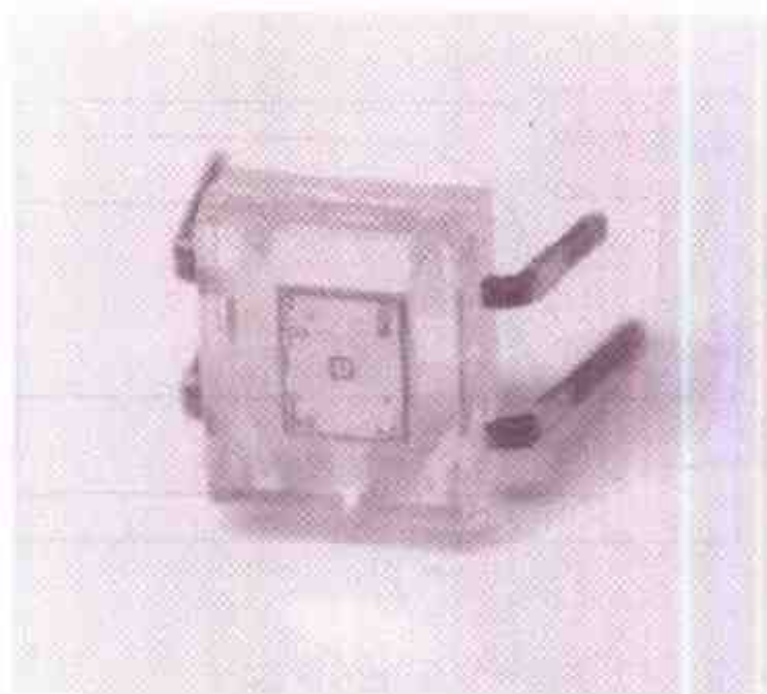
使用调制型的反射型光传感器时,抗干扰光的能力变强,在机器人竞赛中可以安心地使用。在机器人竞赛大会上,为了报导关系的摄像需要明亮的照明情况或者也有从会场窗户射入光的情况,即使是这样的情况,要确认好机器人正常的工作情况。在太阳光的照射中如果机器人工作,无论设定什么样的会场也没有问题。

制作调制型光传感器。图 7.14 是为检测相扑机器人摔跤场的白线作成的传感器电路,使用调制电路内装的光传感器 S4285-40(浜松 Hotniks 制作)。

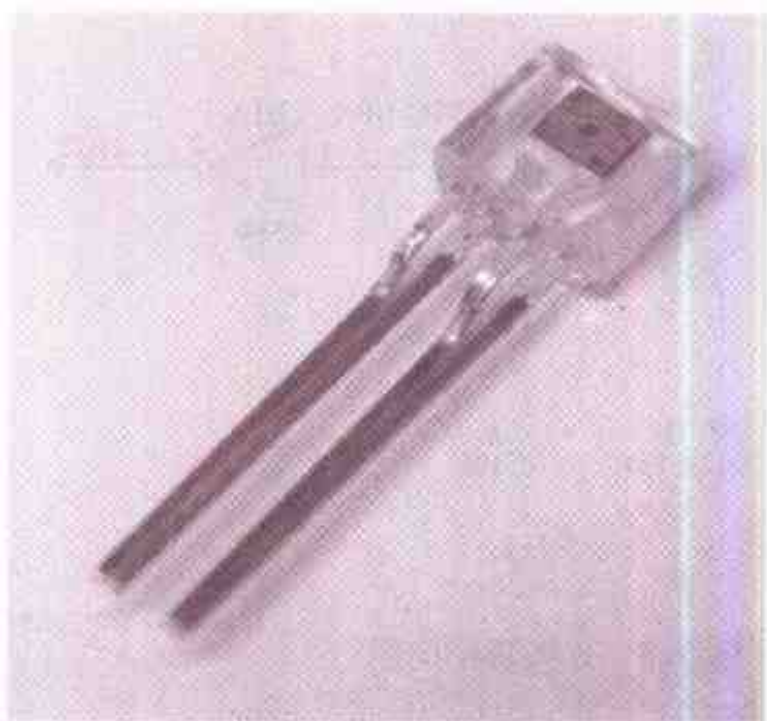
S4285-40 必需是外带红外 LED,而电路非常简单,就是仅安装上监视用 LED 和调整距离用的可变电阻。

在浜松 Hotniks 的调整型 IC 模块中,有 S4282(参见照片 7.10 所示)的倾斜型和超前型(参见照片 7.11 所示)及不同特性的 S7136(参见照片

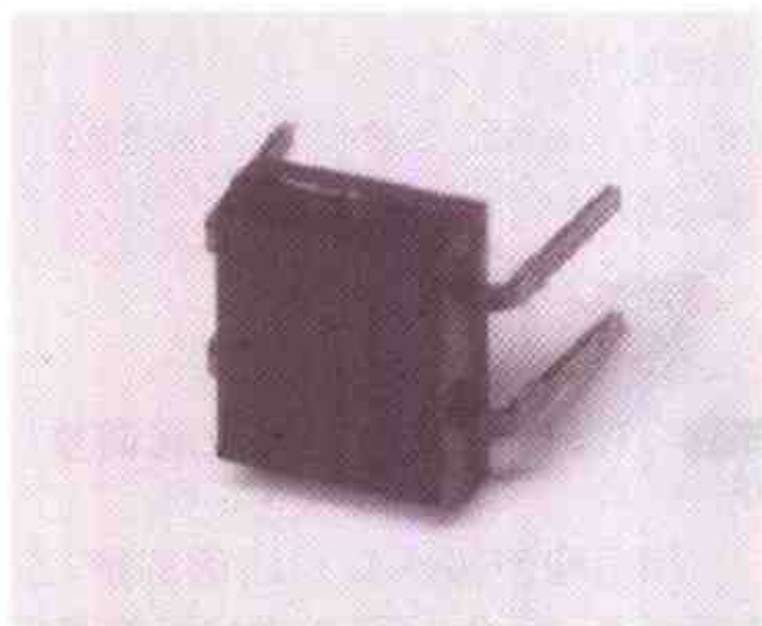
7.12 所示),其感度特性如图 7.11、图 7.12。从使用两方面来看,而采用 S7136 检测距离为好,这从感度特性可知。其中,由于定向性提高,检测的幅度变小。



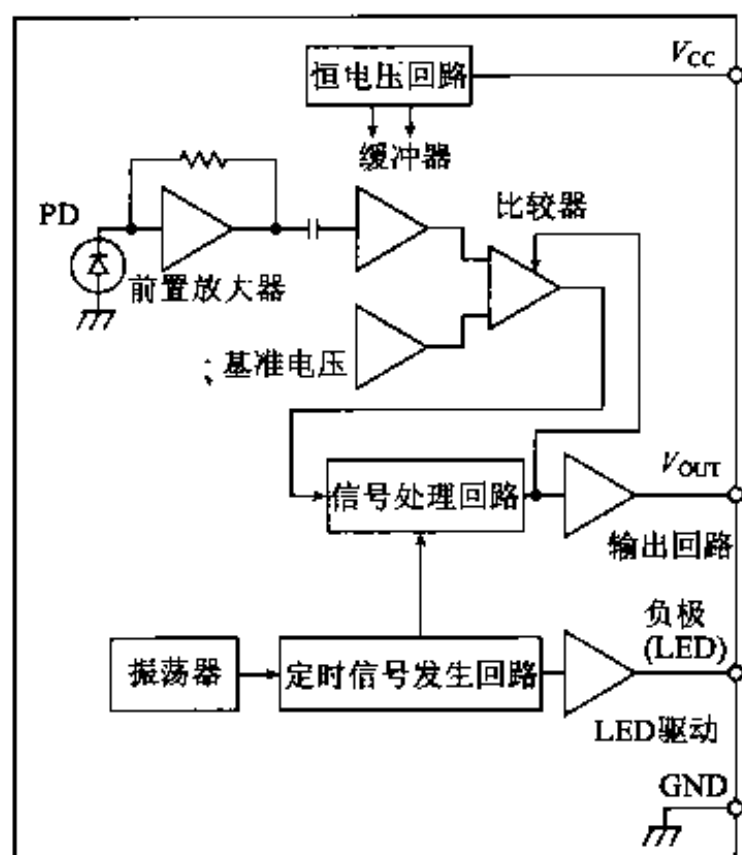
• 照片 7.10 调制型光传感器 S4282-51



• 照片 7.11 超前型光传感器



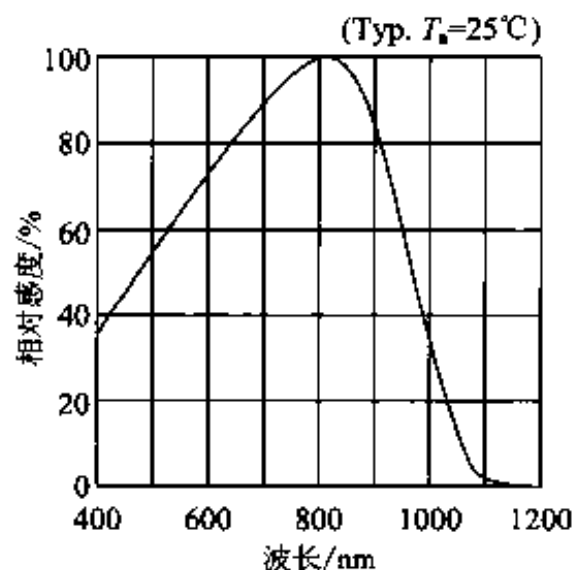
• 照片 7.12 调制型光传感器 S7136



• 图 7.10 调制型光传感器(S7136)的方框图

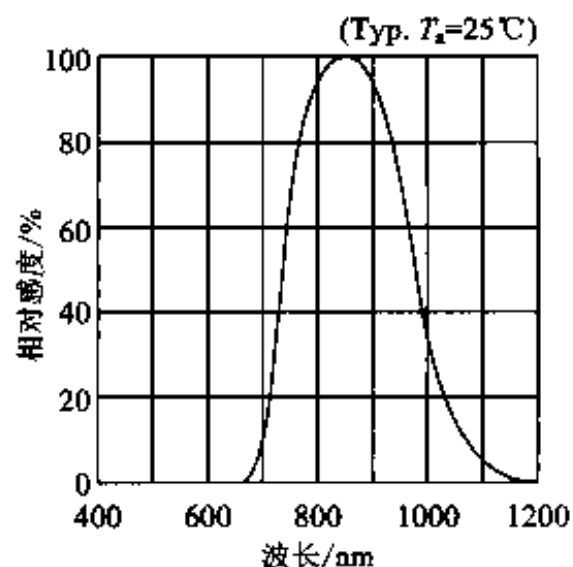
图 7.10 中表示上述两种传感器的方框图,它用前置放大器放大光信号,通过脉冲发光,仅仅读出光的变化部分。用这个定时信号进行比较,为输出回路。加上像这样的脉冲调制,不发光时通过清 0 输出,使其的干扰不受影响。

S6986, S4282-51



• 图 7.11 调制型光传感器(S4282-51)的感度特性

S6846, S7136



• 图 7.12 调制型光传感器(S7136)的感度特性

浜松 Hotniks 株式会社网页
<http://www.hpk.co.jp/>

两者不同仅是感度特性,红外 LED 只要选择发光波长和 S7136 的感度特性一致的即可,这里使用东芝的 TNL109(参见照片 7.13 所示)。

程序

将传感器安装在 J1 上,试制作将传感器的状态表示在 LED 上的程序。

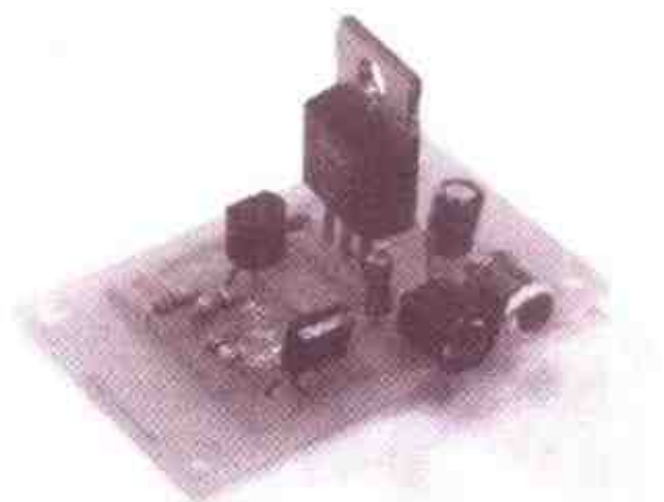
7.5.5 将检测距离变长的方法

为了变长光传感器的检测距离,增大发光量即可。但是作为脉冲发光,平均电流取得十分小。

调制型光传感器 S7136LED 的驱动电流,在检测长距离时候不充足,所以使用外加晶体管,进行电流放大。这里使用 2SC1815 和 2SD560 型晶体管(有关详细情况请参见晶体管规格表 CQ 出版社出售)。另外,LED 安装 2 个 TNL109 二极管,提高了光通量。

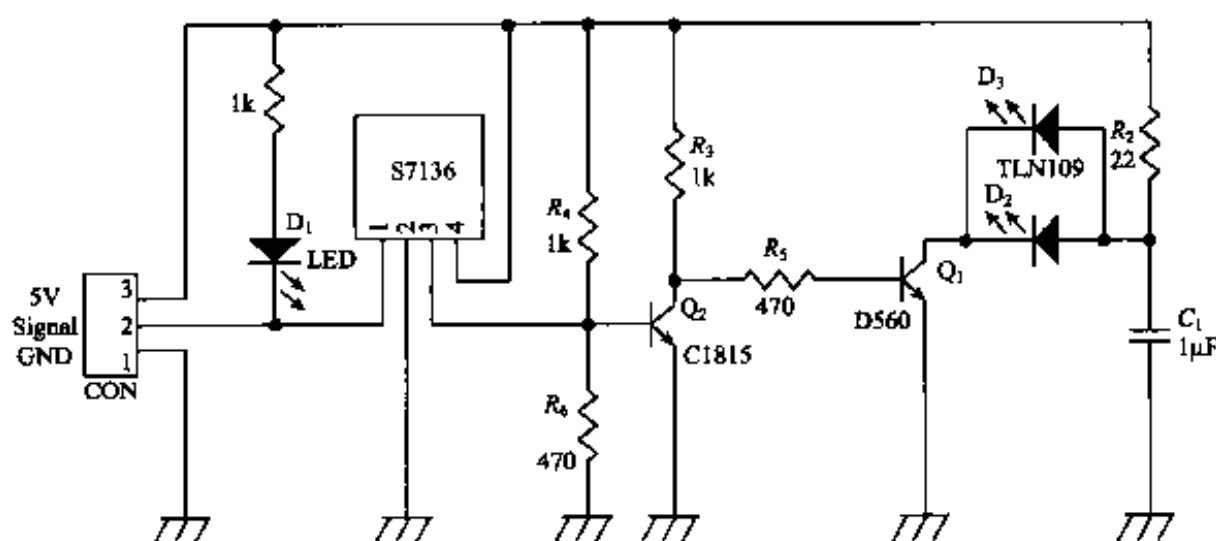
CQ 出版社 1999 年出版最新半导体器件规格表系列,从以下网址可以查找。

<http://www.cqpub.co.jp/hanbai/books/kikaku/99kikaku.htm>



• 照片 7.15 制成的传感器

通过补加这些元件,以前只能检测 10mm 左右距离的传感器,现在却对白色检测物已经能够检测到 200mm 左右。而且,为了提高感度,在接收光元件前面安装透镜即可。在检测实际的相扑机器人的对手时,究竟对手是什么颜色不知道,所以即使是黑色的对手必需确认好其反应距离,这时候对大部分光传感器检测距离变成一半以下(参见图 7.15 所示)。



• 图 7.15 发光量变大的调制型光传感器

7.5.6 读出反射光的强度 测量距离

上述的传感器为 ON-OFF 输出。这里介绍脉冲发光 LED, 并且用光电二极管读出通过在壁上反射光的强度来检测距离传感器。但是壁的颜色必需一定, 否则反射光通量完全改变。

电路图为图 7.16。在 LED 中使用了超高输出的 AlInGaP LED HLMP-EG24, 其红色波长为 626nm。

从以下网页可以得到数据图表

http://www.jpn.hp.com/COMP/products/led_lamps/index.html#HLMP0025

将这个 LED 用晶体管 ON-OFF, 对图 7.16 应该将 PLS 引脚由 CPU 用程序进行 ON-OFF。

从 LED 发出脉冲光, 为了读出由壁面反射返回来光的强度, 使用接收光元件, 作为接收光元件使用了光电

二极管, 其为东芝的 TPS703。

从以下网页可以获得数据图表

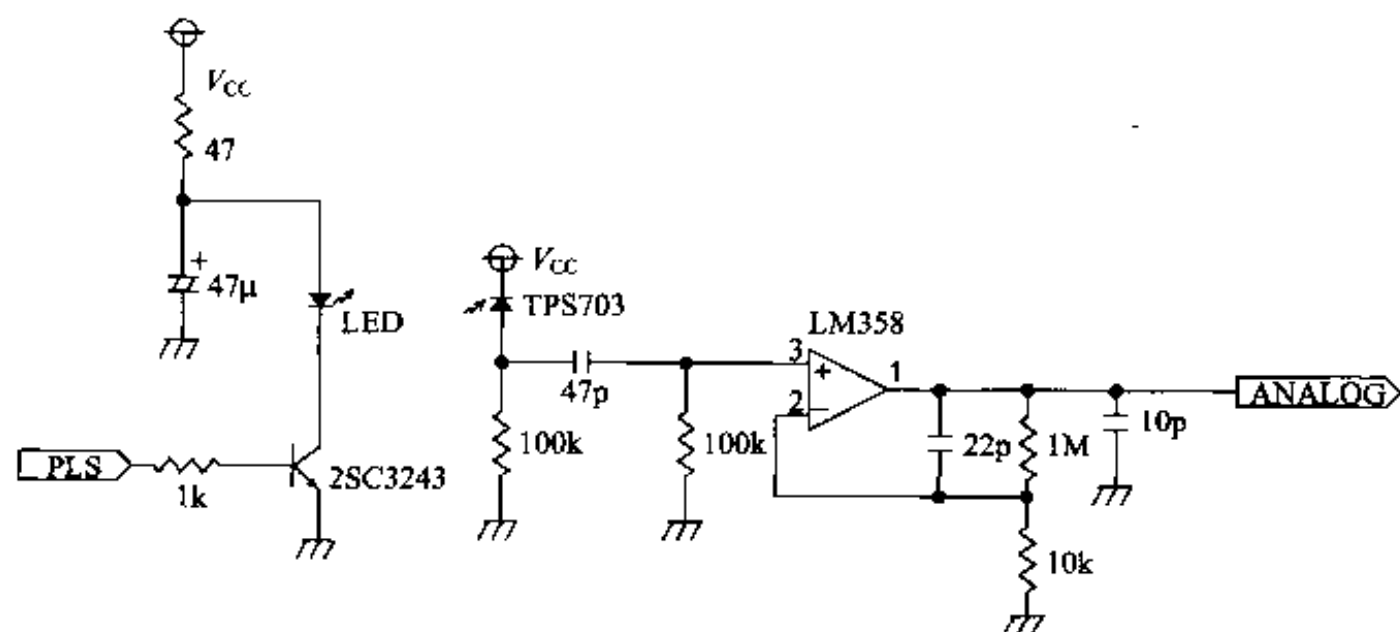
<http://www.semicon.toshiba.co.jp/noseek/jp/td/14frame.htm>

用运算放大器, 放大光电二极管的输出。在运算放大器中使用了 LM358。

由以下网页可以得到数据图表

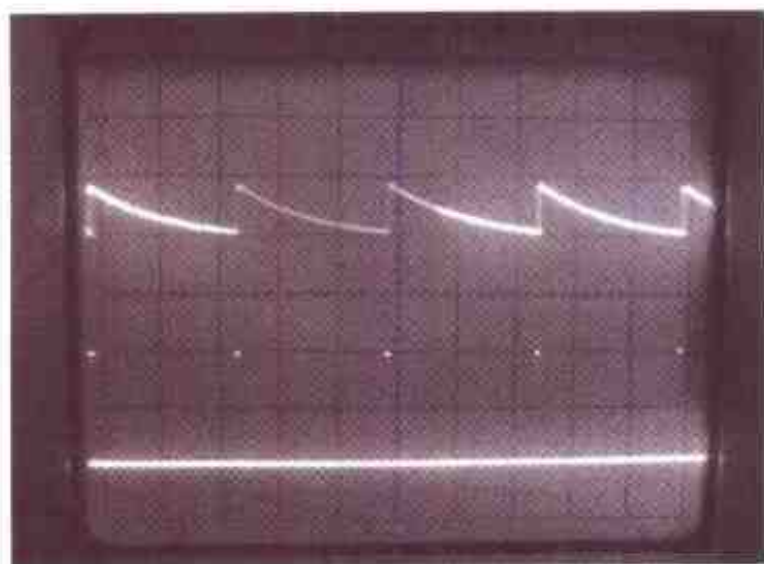
<http://www.national.com/pf/LM/LM358.html#Datasheet>

将运算放大器的输出如果在瞬间进行 AD 转换, 可以读出反射光的强度。只要将图 7.16 的模拟引脚连接在微型计算机的 AD 转换器输入端口即可。照片 7.16 为将脉冲输入和模拟输出在示波器输出的波形。就是在这个模拟输出峰值附近进行 AD 转换。



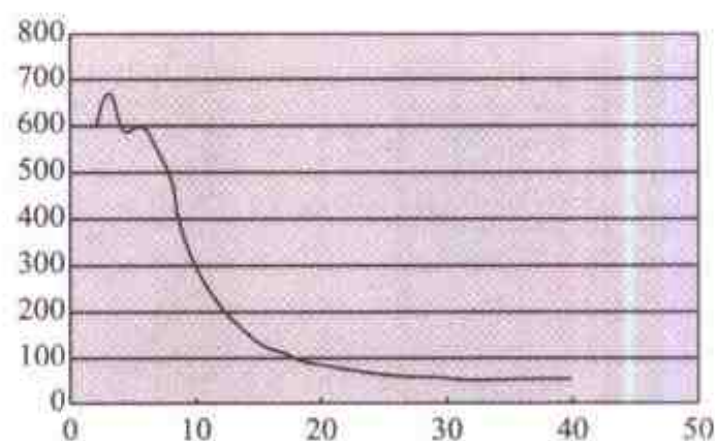
• 图 7.16 光传感器电路

图 7.17 是对距离输出的曲线,随着距离变远,输出降低。这里采用对数坐标,将其加以变换变为图 7.18 的形式。用这个变换可以获得大概的线性值。

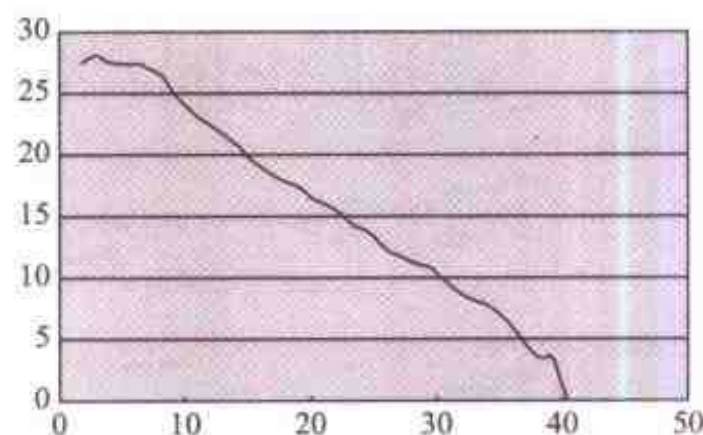


• 照片 7.16 输入脉冲和模拟输出波形

为了使传感器数增加,只要安装多个上述电路即可,然而多数使用 AD 端口或输入/输出端口,所以如图 7.19 使用模拟开关转换时,可以使用 AD 端口一个 4 英寸的传感器。

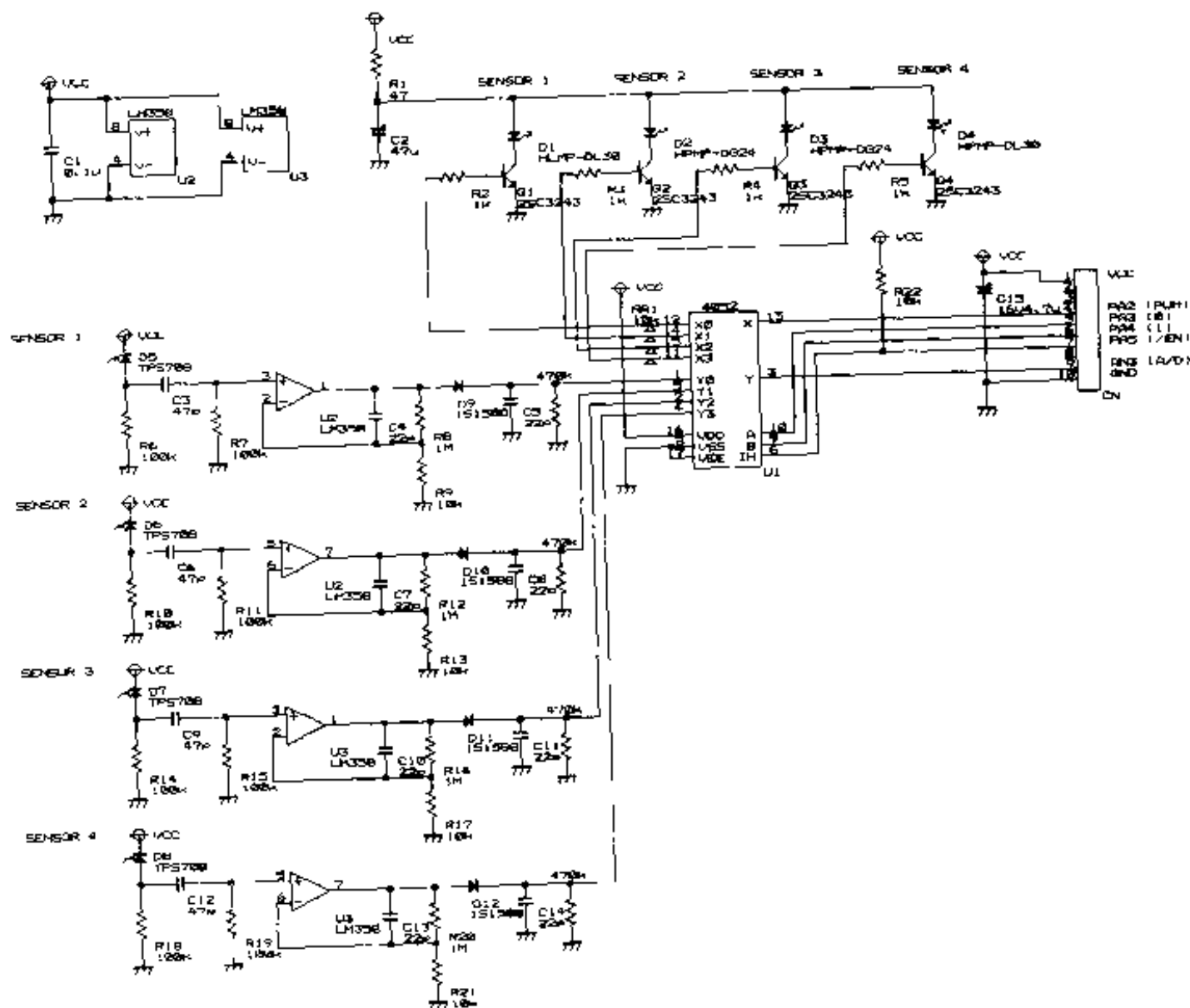


• 图 7.17 AD 值(对数坐标)



• 图 7.18 AD 的处理值

在模拟开关中使用 4052,模拟开关为可以转换模拟量的 IC 模块。



• 图 7.19 用模拟开关转换 4 通道电路

7.6 超声波传感器

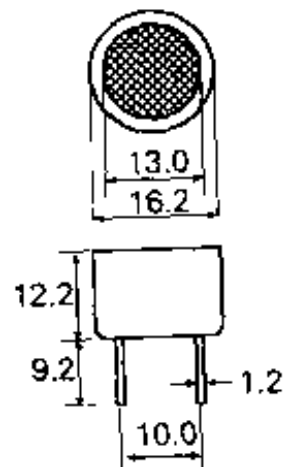
7.6.1 超声波传感器

超声波传感器由压电陶瓷构成，由发送器和接收器组成。

压电元件施加力时产生电压，它有施加电压时产生变形的特性，具有和压电蜂鸣器同样的特性。利用这个特性压电陶瓷元件为超声波传感器。

超声波传感器在发信号侧使用 T40-16，在接收信号侧使用 R40-16，图

7.21 中表示它的外形尺寸，表 7.2、表 7.3 中表示各种各样的规格型号。



• 图 7.21 T40-16、R40-16 的外形尺寸

表 7.2 规格表(发送器)

项 目	T40-16	T40-18A	T40-24A
中心频率(kHz)	40±1	40±1	40±1
声压级(中心频率) (0dB=0.0002μbar)	115 dBMin	115 dBMin	
频带宽(kHz/100dB)	6.0Min(103 dB)	6.0Min(100 dB)	6.0Min(103 dB)
静电容量(pF)	2400±25%	2400±25%	2400±25%

表 7.3 规格表(接收器)

项 目	R40-16	R40-18A	R40-24A
中心频率(kHz)	40±1	40±1	40±1
感度特性(中心频率) dB/V/μbar	-64Min	-64Min	-64Min
频带宽(kHz/100dB)	6.0Min(-71 dB)	6.0Min(-71 dB)	6.0Min(-71 dB)
静电容量(pF)	2400±25%	2400±25%	2400±25%

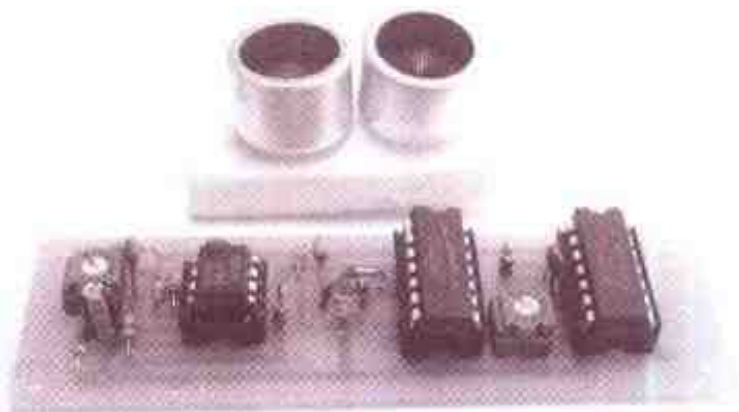
超声波传感器是日本陶瓷社的产品
<http://www.nicera.co.jp/>

发信号侧由 40kHz 的通信电路、控制 ON-OFF 及驱动器组成。接收信号侧由放大、比较器和积分电路组成。图 7.20 中表示超声波传感器电路。超声波传感器为了容易反射声波,对测定物体最好成直角安装。

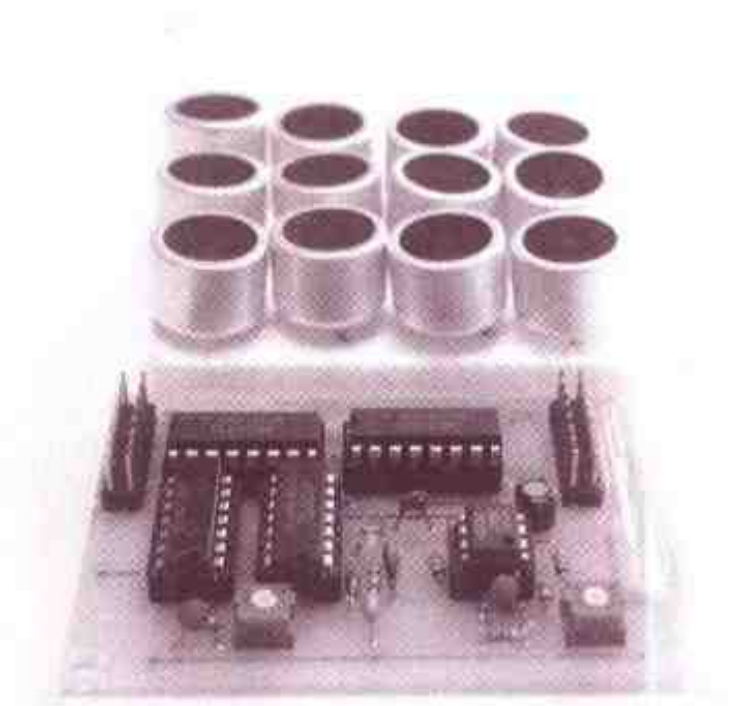
传感器的调整通过图 7.20 的可变电阻R₃变化振荡频率,所以使其与发送器的谐振点一致。用可变电阻R₅改变比较器的比较电压,由此可以调整感度。

7.6.2 多通道超声波传感器

图 7.22 为使用 8 通道转换的模拟开关,根据 3 位的信号、在每个通道上可以读入信号的电路。因此,发送电路及接受电路和图 7.20 相同,分别用一个电路即可。



• 照片 7.17 超声波传感器



• 照片 7.18 多通道超声波传感器

程序

对于程序在 ABC 引脚上从 000B 到 111B 二进制数的输出值对应各个通道。

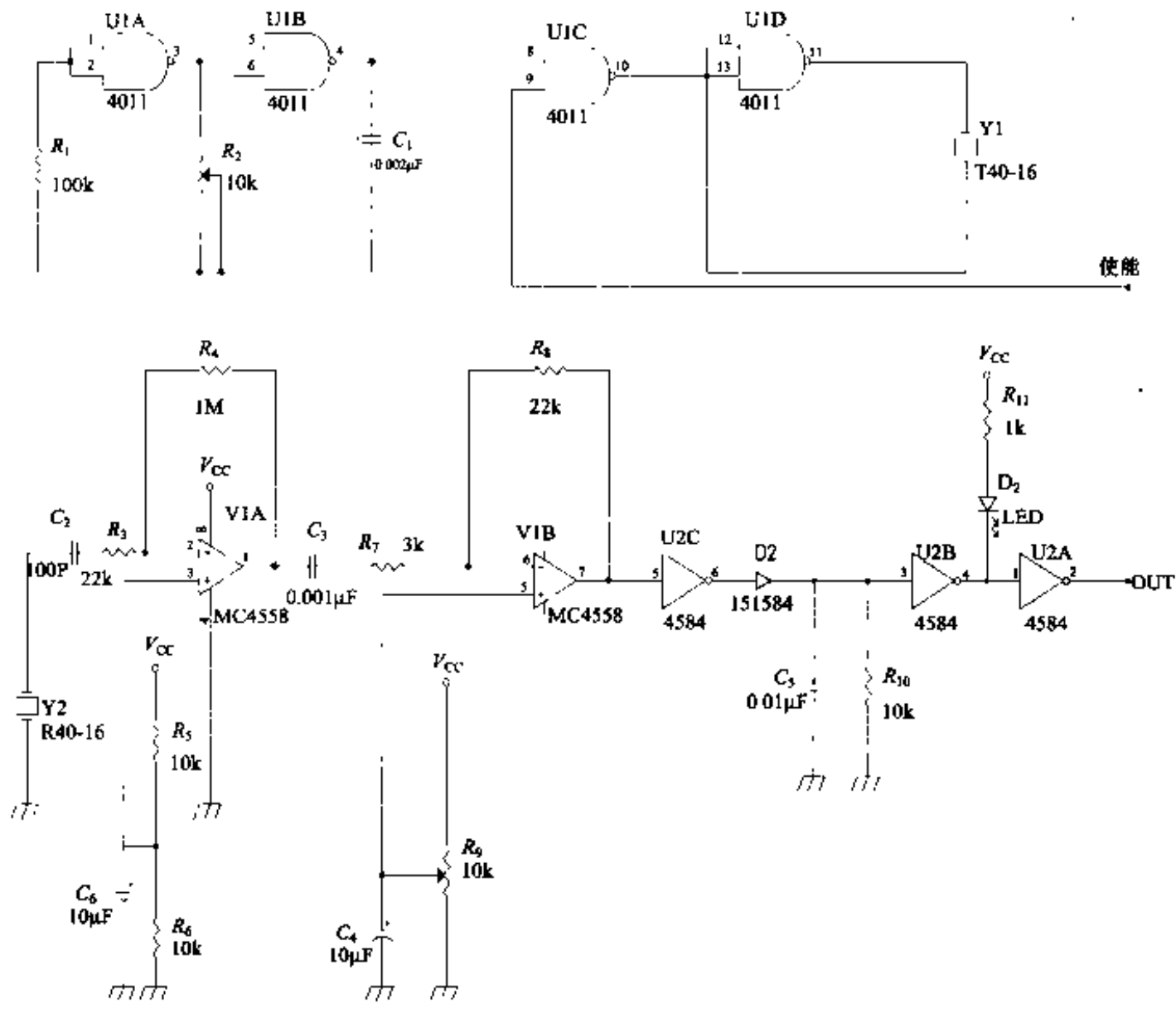
读入 0 通道信号时,ABC 引脚上输出 000B,使能(Enable)引脚变为高电平。这里接收电路的输出只要能读 ON 或 OFF 即可,这时,反射波以声速互通信息,所以必需等待这个时间以

上。同样,改变 ABC 序号,只要能读入全部的通道即可。

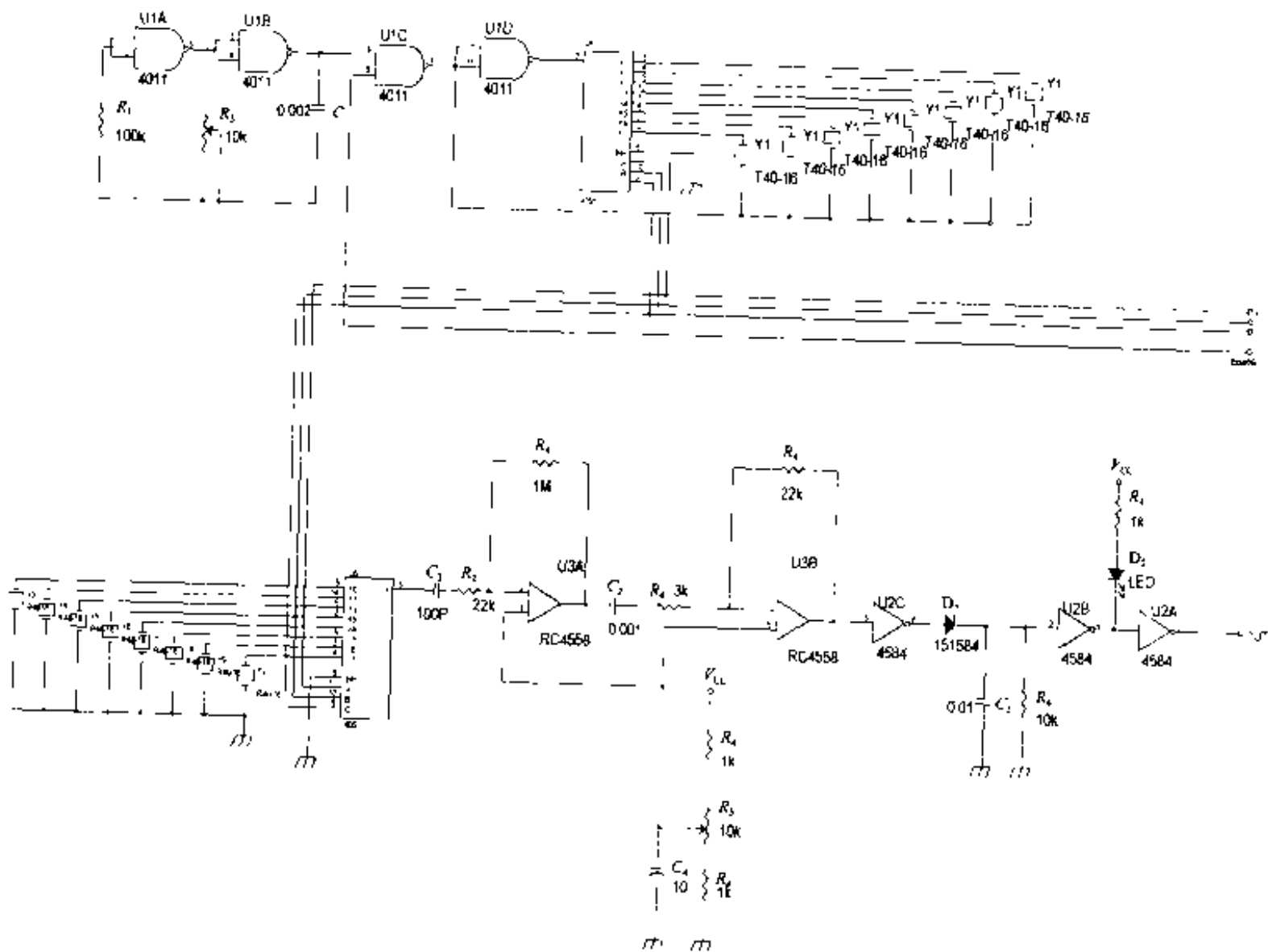
读距离时将 ABC 引脚设置为一组后,使能(Enable)引脚变为高电平。测量从此时到接收信号变高电平的时间 t 。由于超声波以声速 VS 互通信息,所以距离 L 用下式即可以求出:

$$L = VS \text{ (m/s)} \cdot t \text{ (s)} / 2$$

声速在常温(20℃)为 340 m/s。



• 图 7.20 超声波传感器电路

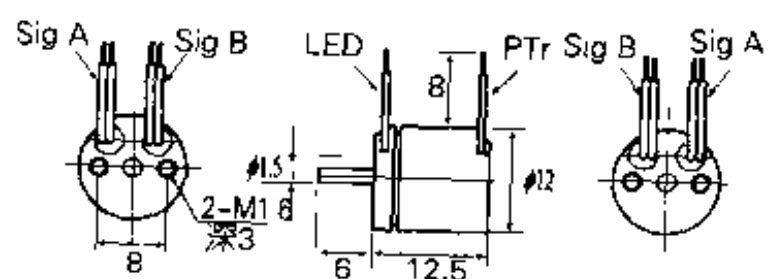


• 图 7.22 多通道超声波传感器电路

7.7 旋转编码器(光电编码器)

旋转编码器就是用光传感器测量旋转的切口数。其中有 A 相和 B 相 2 个光传感器。并且,2 个传感器相位错开一定角度配置。

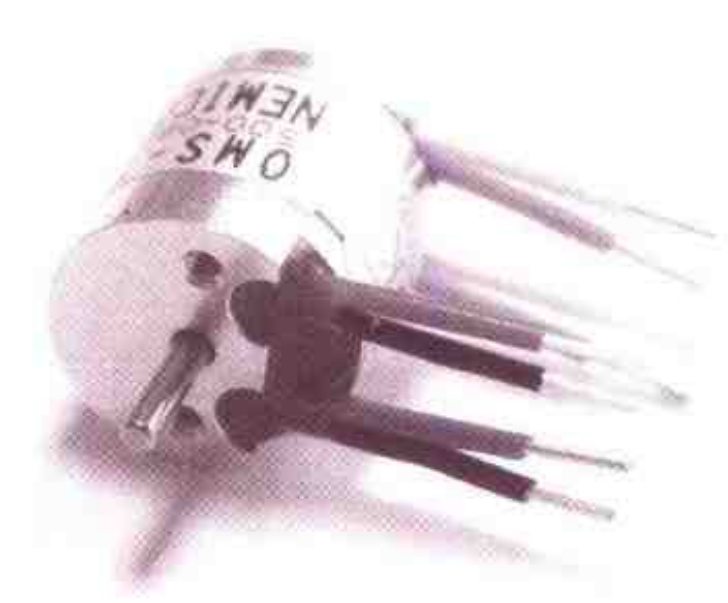
这个旋转编码器是用来测量机器人移动距离或速度的。这里,使用日本电产 Nemicon 株式会社 OMS-100 的外径 $\phi 12$ 超小型编码器。每转输出 100 个脉冲,外形尺寸在图 7.23 中表示,形状在照片 7.19 中表示。另外,内部回路为图 7.24,LED 和光电晶体管每 2 个内装起来。



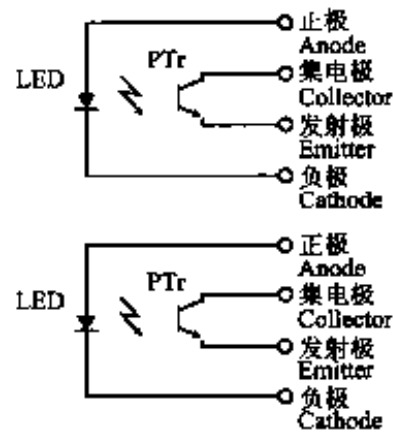
• 图 7.23 OMS-100 编码器的外形尺寸

有关这种规格的详细说明,请参照下述网页。

<http://www.nemicon.co.jp/ie/home.html>



• 照片 7.19 OMS-100 编码器

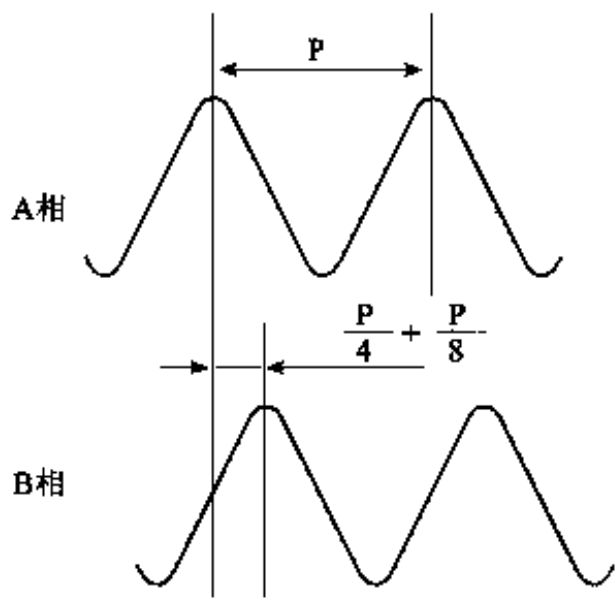


• 图 7.24 内部回路

由于是光传感器,所以和图 7.9 的电路相同即可。传感器封装在中间,由于不担心受干扰光的影响,所以不需要调制回路。输出波形变为图 7.25 的波形,在光电晶体管输出的后面安装逆变器,构成输出矩形波的电路。电路图为图 7.26 所示,旋转编码器的连接如表 7.4 所示。另外,电气特性在表 7.5 中表示。

表 7.4 OMS-100 的连接

颜色	连接
红	正极
黑	负极
蓝	集电极
白	发射极



• 图 7.25 输出波形

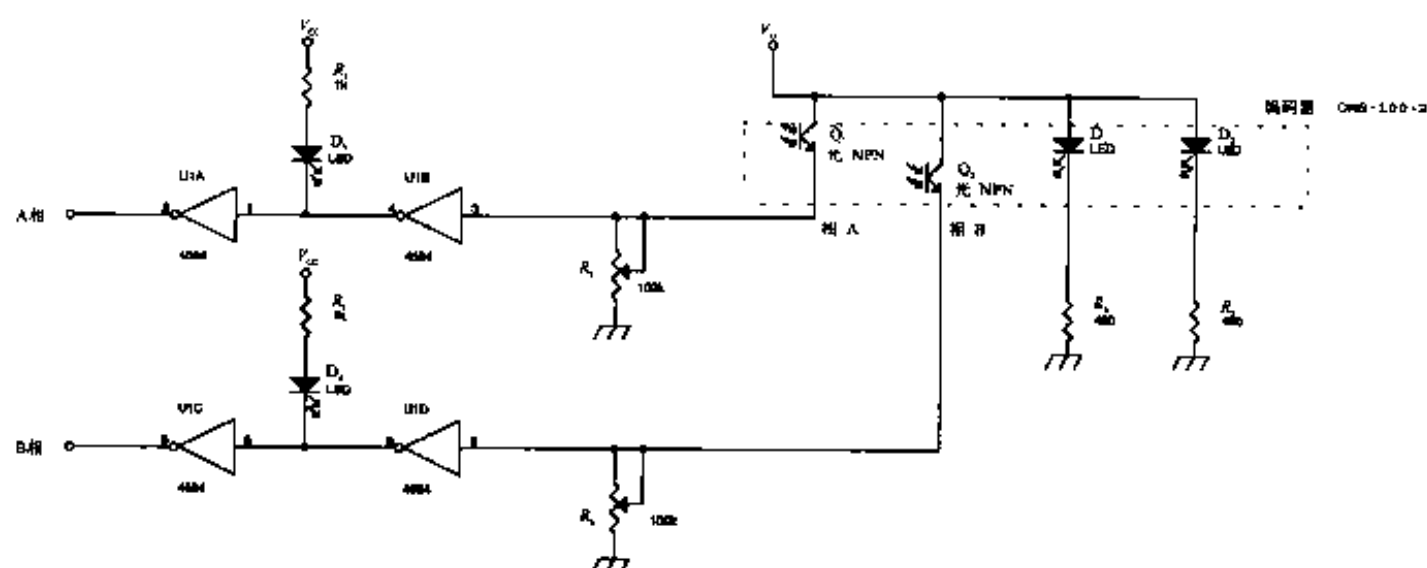
表 7.5 OMS-100 的电气特性

集电极、发射极间电压	$V_{CE0}: 20V$
发射极、集电极间电压	$V_{ED0}: 5V$
集电极电流	$I_C: 20mA$
集电极损耗	$P_C: 75mW$
允许损耗降低率(25~70℃)	$-1.0mW/℃$

相扑机器人,是在旋转编码器轴上,安装照片 7.20choler Q 的轮胎(参见照片 7.21 所示)。而且,将其安装在如照片 7.22 机器人的下面即可。如果这样,不管电动机吭吭旋转,旋转编码器没有运动,就可以判断轮胎在打滑,即对手没有运动。这时候,可以



• 照片 7.20 CholerQ

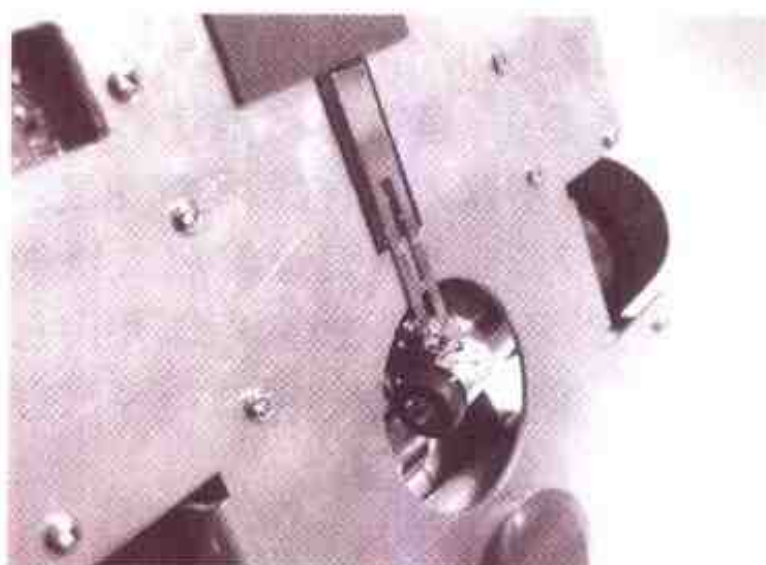


• 图 7.26 旋转编码器电路

书写再一次返回正面接触的程序。另外,虽然电动机一定前进,但是旋转编码器最好是避开返回的情况。



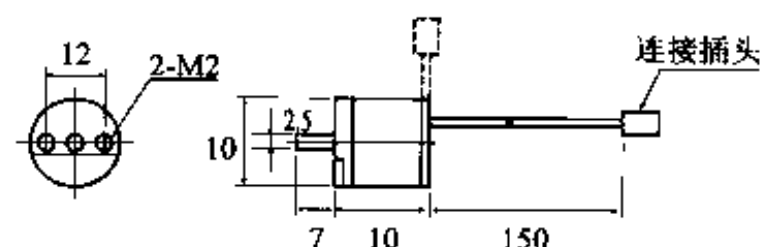
• 照片 7.21 安装轮胎的地方



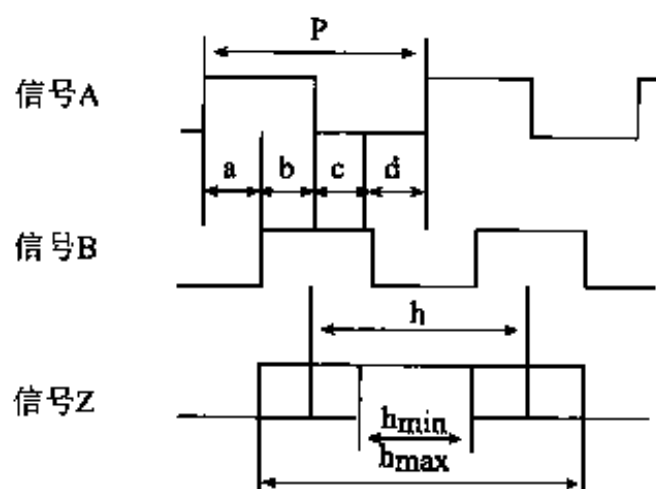
• 照片 7.22 安装在机器人上的情况

最近,也有方形波输出旋转编码器,可以直接连接在 CPU 插件板上。图 7.27 是日本电产 Nemicon 株式会社内装波形整形回路的旋转编码器 OME-A。图 7.28 的方形波可以通过连接器直接输出。

<http://www.nemicon.co.jp/ie/home.html>



• 图 7.27 OME-A 外形尺寸



• 图 7.28 方形波输出波形

程序

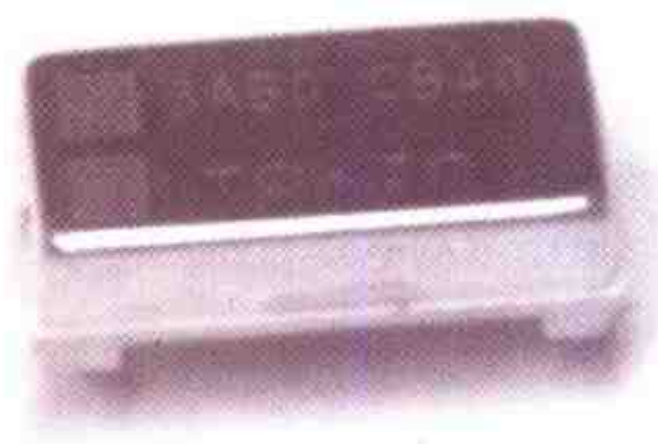
对 H8 旋转编码器读入简单,只要

设定计相方式就可以定时地输入数值。试进行角度、速度和距离的检测。

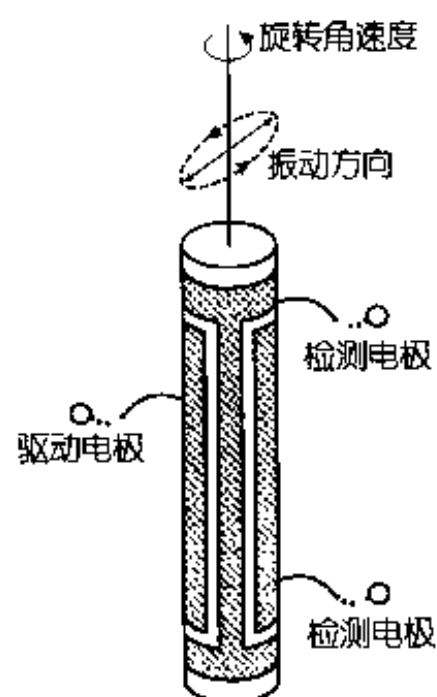
7.8 回转仪传感器

回转仪传感器是检测旋转角速度的传感器,机器人如何旋转,只要积分角速度就可以得到。

照片 7.23 是使用 Tokin 制作的陶瓷振子的回转仪传感器。村田制作所也正在出售。原理就是如图 7.29 在使其振动的陶瓷振子上,检测由旋转施加的力振动的变化。



• 照片 7.23 Tokin 制作的 CG-L33

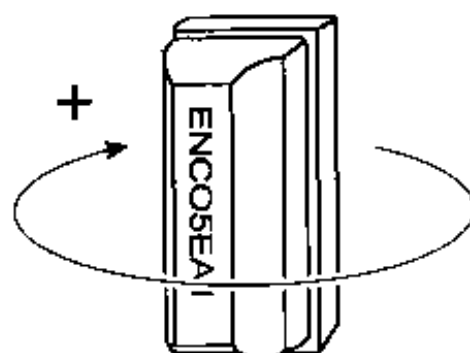


• 图 7.29 回转仪传感器的构造

村田制作所

<http://www.ijnet.or.jp/murata/products/japanese/catalog/catalog.html> Tokin
<http://tokin.co.jp/>

对于传感器,可以检测如图 7.30 那样旋转方向的角速度,所以必须做到安装方向正确。



• 图 7.30 传感器形状

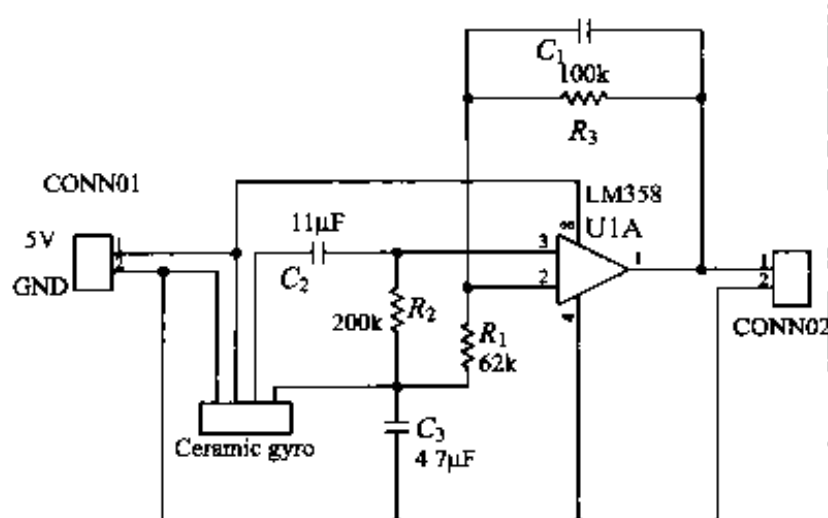
如图 7.31 所示,用运算放大器放大传感器输出,通过在微型计算机上的 AD 转换器读入,以检测角的速度。

照片 7.24 是带有运算放大器的传感器,市场上正在出售,所以使用它也可以。这张照片就是共立电子产业株式会社出售的产品。

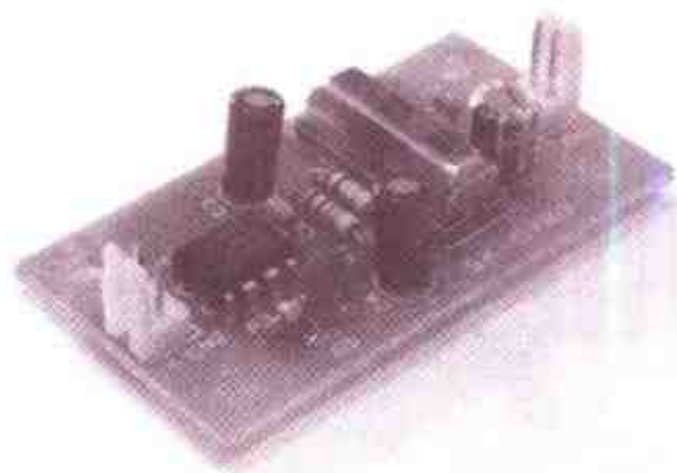
程序

将这个传感器的输出连接在微型计算机的 ADC 上,在个人电脑上试表

示其角速度。



• 图 7.31 传感器电路



• 照片 7.24 带有运算放大器的回转仪传感器

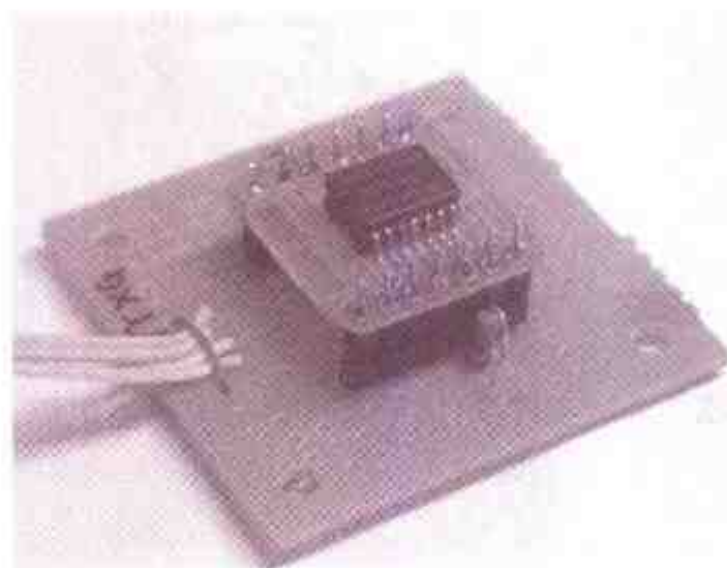
共立电子产业株式会社

<http://www.kyohritsu.com/>

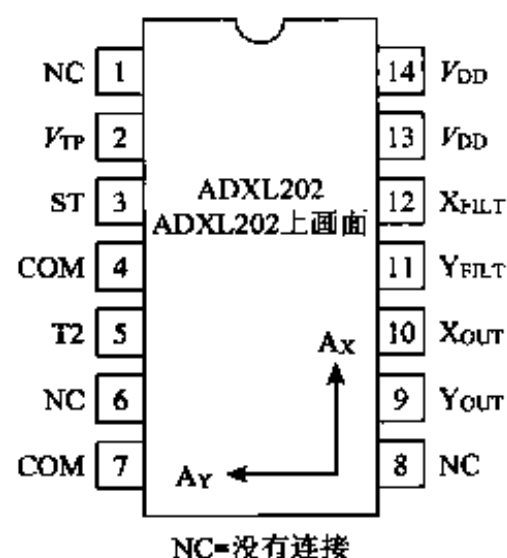
7.9 G 传感器

G 传感器是为了测量重力加速度的传感器。ADXL202 是模拟仪器社制作的 2 轴重力加速度传感器。在图 7.32 中表示其引脚配置图。对于利用变形测量仪的这个传感器芯片上,如图 7.33 内装放大器或温度保证回路

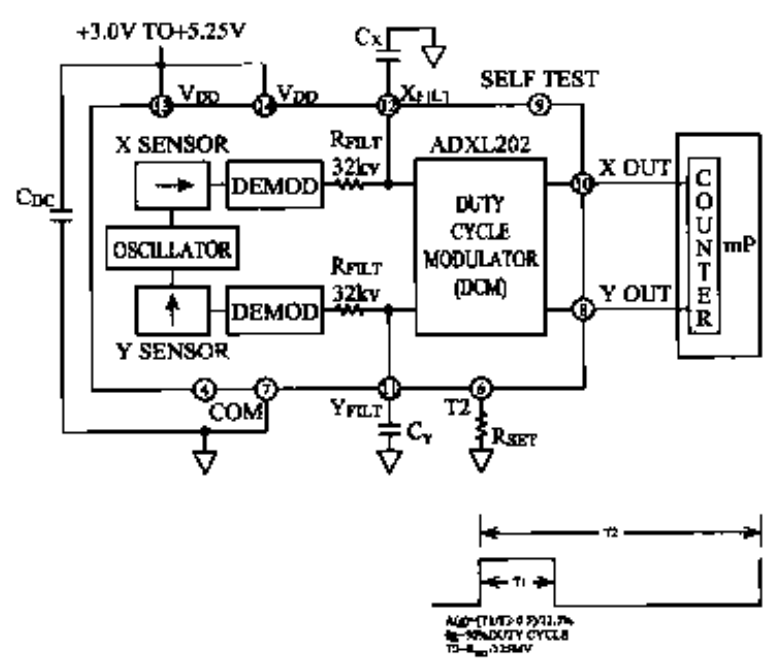
等。另外,传感器输出为数字信号,其占空度(对周期脉冲宽度的比率)与 2 个检测轴各自的加速度成正比例输出。其输出周期用电阻(R_{SET}) 在 0.5 ms 到 10ms 的范围内可以调整。



• 照片 7.25 制成的传感器插件板



• 图 7.32 管脚连接图



• 图 7.33 内部方框图及附加回路

由模拟仪器社网页,可以得到传感器数据图表。

<http://www.analog.com/>

7.10 对人反应的传感器

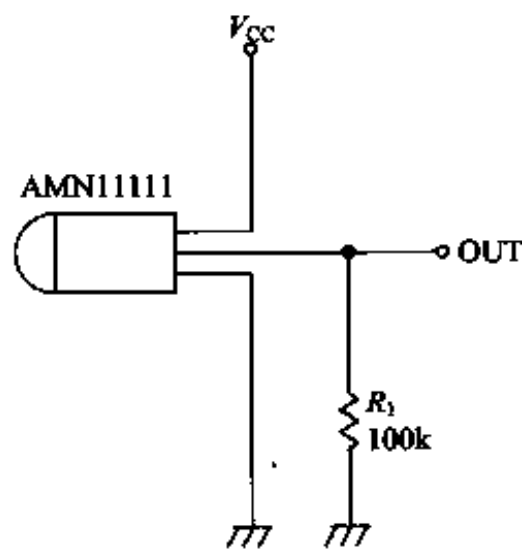
对人反应的传感器,多用于自动门或便所自动水洗等。安装这种传感器的机器人可以判断人。

这种传感器使用热电元件,就是从人体 $10\mu\text{m}$ 附近发出红外线反应。内部具有 2 个红外传感器,消除背景变化连接使用。为此,仅人活动时反应,能够输出。更好地理解这个性质必需进行程序设计。

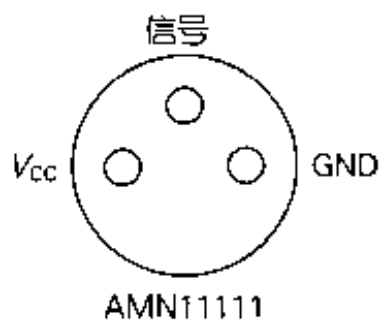


• 照片 7.26 AMN11111 运动传感器

照片 7.26 是松下电工株式会社的运动传感器 AMN11111。内部装好放大器,如图 7.34 所示,仅用补加电阻就能输出。信号线的连接为图 7.35 所示。



• 图 7.34 带有附加电阻的运动传感器电路



• 图 7.35 信号线的连接

程序

可以判断人的活动,在人活动时 LED 闪烁 1 分钟。在这段时间内如果人活动,计数器将复位(置“0”),之后再 进行 1 分钟 LED 的闪烁。

松下电工株式会社

<http://www.napion.com/>

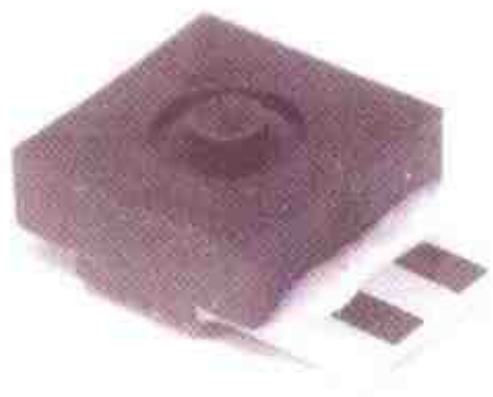
7.11 感压传感器

这里使用 Inaba 橡胶株式会社的感压导电性合成橡胶传感器。

Inaba 橡胶株式会社

<http://www.inaba-rubber.co.jp/>

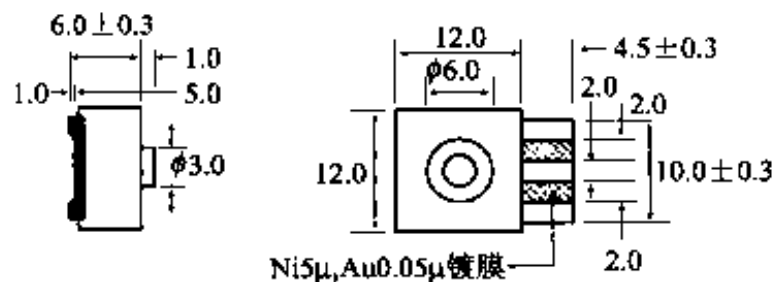
这种传感器使用合成橡胶,根据压力的强弱连续地变化电阻值。



• 照片 7.27 传感器的照片

这种传感器的照片在照片 7.27 中表示,外形尺寸在图 7.36 中表示。

这种传感器的特性如表 7.6 中所示,根据施加力的不同电阻从 500k Ω 到 500 Ω 发生变化。如果察觉这个电阻变化,就可以测量压力或力。



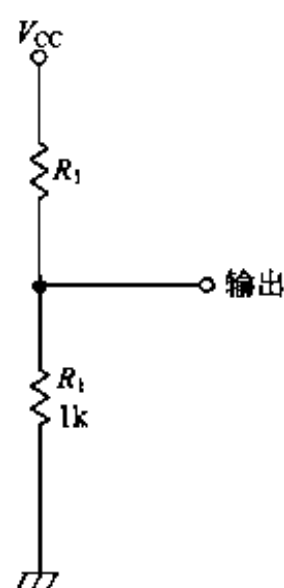
• 图 7.36 传感器外形尺寸

表 7.6 传感器的特性

最大允许电压	30V
推荐使用电压	3~6V
最大允许电流	20mA
推荐使用电流	5mA
无负载电阻值	500k Ω 以上
最大负载电阻值	500 Ω
最大允许负载	4.0kgf($\approx 4 \times 9.8N$)

这个传感器的连接电路如图 7.37 所示。串联连接 1 k Ω 左右的电阻,从其连接部分读出输出,将其读入 AD 转

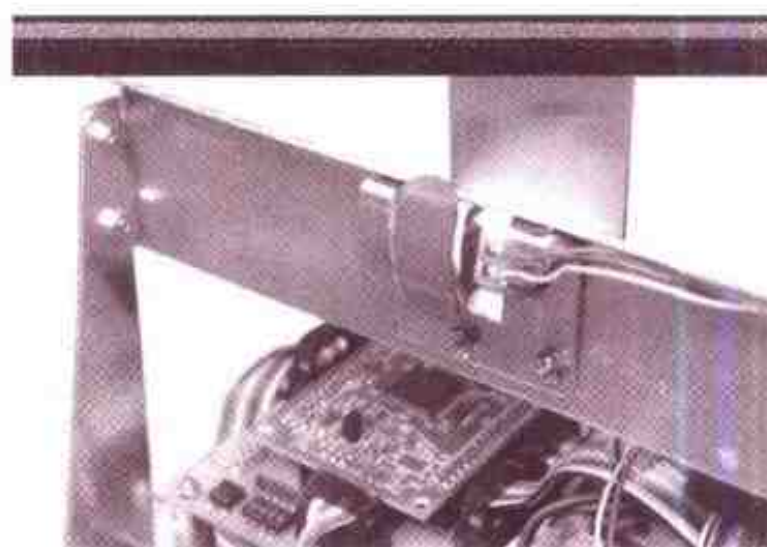
换器,只要能读取电压即可。



• 图 7.37 传感器电路

在相扑机器人中,可以使用它判断是否碰到对手前面伸出的臂这个障

碍物。如照片 7.28 所示,用两面胶带粘接臂的下面,用电话卡引导。而且为了容易滑动将贴上家具导轨固定即可。



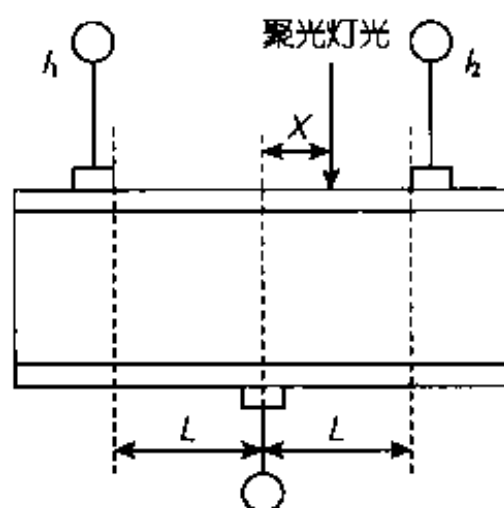
• 照片 7.28 安装感压传感器

7.12 PSD 距离传感器

对机器人,准确地知道距离是很重要的,而且要提高它的精度,即使在控制性方面也具有重要的意义。在 7.5 节中叙述了对检测反射光通量的大小方式,对手的颜色或反射率不同时,检测距离精度变坏。因此,这里着眼利用三角测量原理的 PSD 距离传感器。

1) PSD

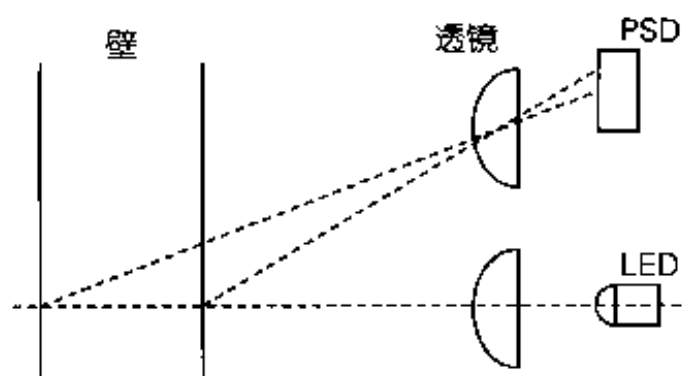
在图 7.38 中,聚光灯光接触到位置 X 与各自流过的电流 I_1/I_2 成正比例,由这个电流差可以判断在任何点是否接触到聚光灯,这是 PSD 的原理。



• 图 7.38 PSD 原理

2) 三角测量原理

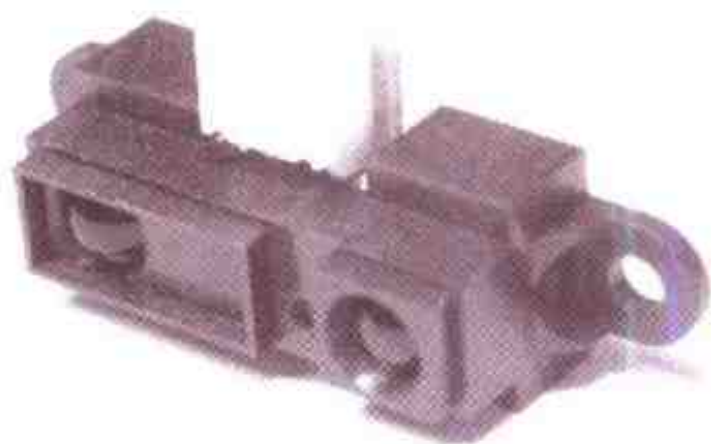
如图 7.39 臂的位置变化时,在 PSD 上产生光的聚光灯位置变化,如果利用它,可以计算到臂的距离。



• 图 7.39 三角测量原理

3) PSD 距离传感器(夏普的 PSD)

单个 PSD 在浜松 Hotniks 社等出售,而光学系统或输出回路制作困难,所以使用照片 7.29 夏普 GP2D12 时简单。图 7.40 是其外形形状尺寸,内装有透镜及模拟输出回路,将输出如果在 AD 转换器上输入,就可以读出距离。最大距离可以测量到 80cm,这时的电气特性在表 7.7 中示出。



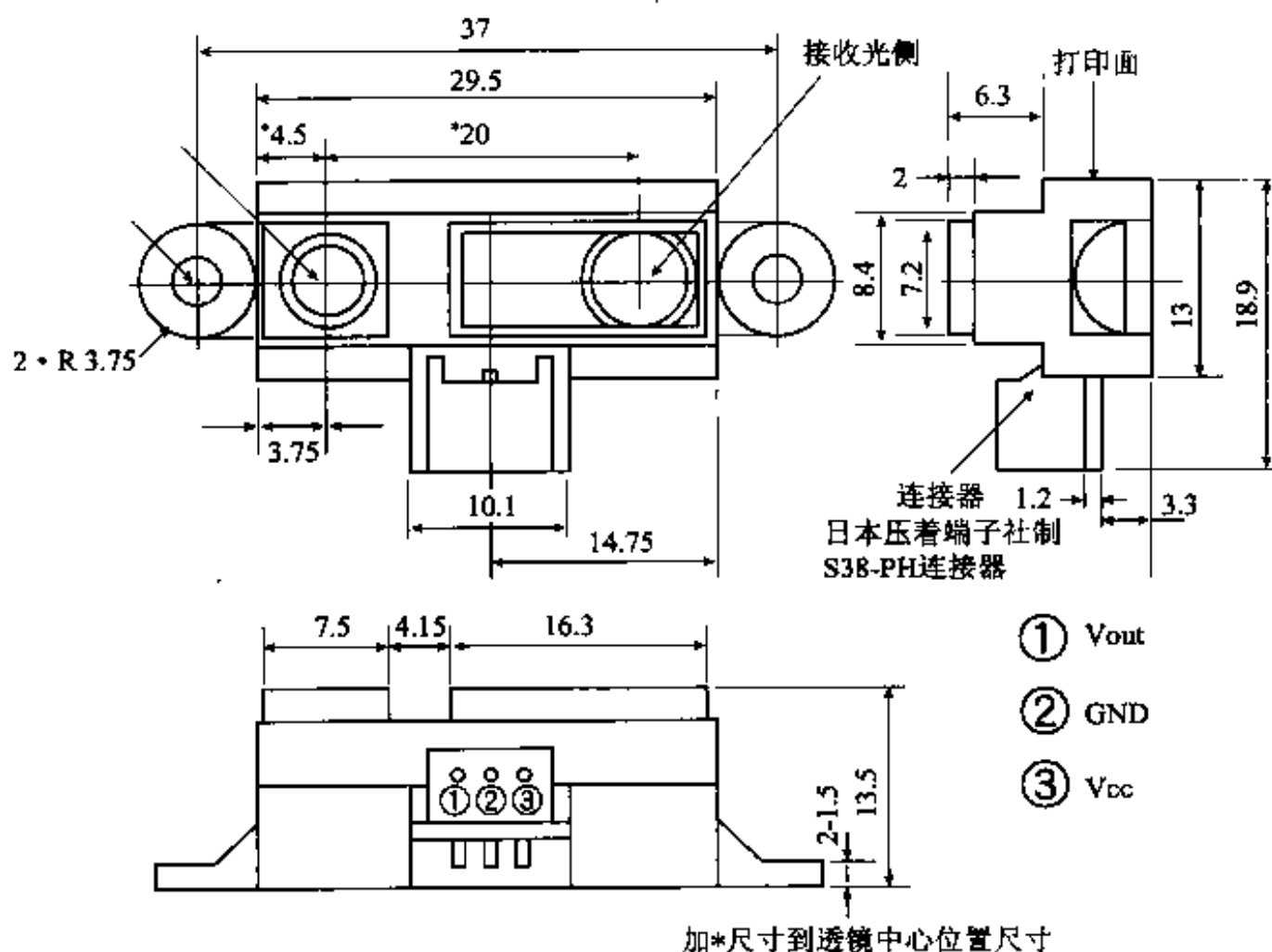
• 照片 7.29 PSD 距离传感器

从以下网页可以得到数据图表

<http://www.sharp.co.jp/ecg/Application/tv.html>

程序

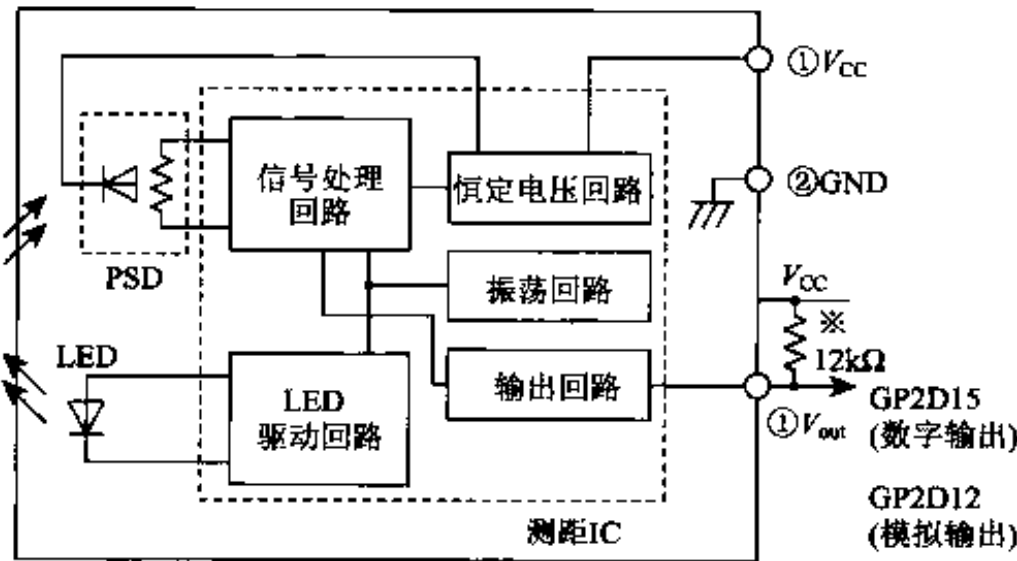
将传感器输出连接在 J2 上,从 AD 转换器读取数据,算出距离,并且表示在显示器上。



• 图 7.40 PSD 距离传感器外形形状尺寸

表 7.7 GP2D12 电气的光学特性 (T_a = 25℃、V_{cc} = 5V)

项 目	符号	条 件	最小值	标准值	最大值	单位
测量范围	ΔL	—	10	—	80	cm
输出端电压	V_o	$L = 80\text{cm}$	0.25	0.4	0.55	V
输出电压差	ΔV_o	L 变化量(从 80cm~10cm)	1.75	2.0	2.25	V
平均消耗电流	I_{cc}	$L = 80\text{cm}$	—	33	50	mA

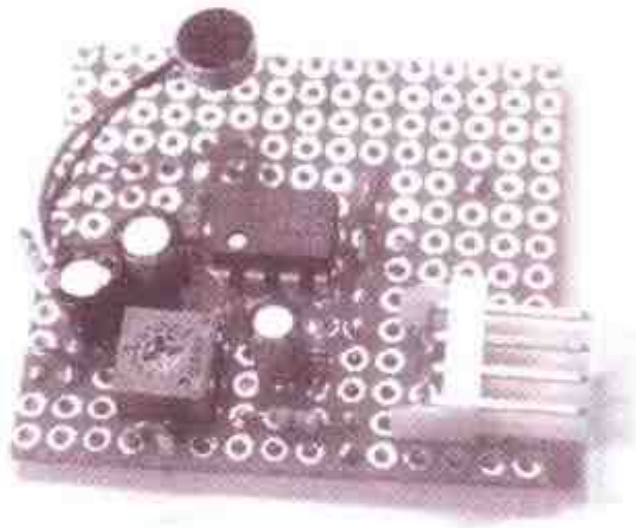


※拉起电阻12kΩ,仅GP2D15需要

• 图 7.41 内部方框图

7.13 在 AD 转换器上输入声音

最近声音识别技术正在提高。但是,在 H8 微型计算机上进行时,给 CPU 增加的负担过大。这里,制作能进行判断音量的扬声器,用它来进行输入声音的音量是大,还是小程度的判断。在声音输入时,使用电容麦克风(如照片 7.31 所示)。将声音用运算放大器(如照片 7.30 所示)放大,运算放大器作为交流放大器使用,其电路图如图 7.42 所示。



• 照片 7.30 微型放大器



• 照片 7.31 电容微音器

传感器厂家的网址

· 旋转编码器

Nemicon

<http://www.nemicon.com/ie/home.html>

· 光传感器

· Everlight(中国台湾的公司)

<http://www.everlight.com/index.cfm>

夏普

<http://www.sharp.co.jp/ecg/New Products/>

Rohm

http://www.rohm.com/index_f.html

新光电子

<http://www.shinkoh-elecs.com/JAPCON.html>

东芝

<http://www.semicon.toshiba.co.jp/noseek/jp/td/14frame.htm>

株式会社 Tokin

<http://www.tokin.co.jp>

村田制作所

<http://www.lijnet.or.jp/murata/products/japanese/catalog/catalog.html>

浜松 Hotniks

<http://www.hpk.co.jp/>

chapter

8

机器人的手和脚

机器人竞赛中使用直流电动机，用其构成机器人的手、脚部分。在本章中，就电动机的工作原理、电动机驱动器的制作方法，以及有关其控制方法等进行阐述。

8.1 电动机

构成机器人手和脚的基本部分是驱动系统。在驱动时,目前使用的是电动机,所以就有关电动机进行讲解。在电动机中有步进电动机、交流电动机等各种形式的电动机,然而,这里就有关一般在机器人竞赛中使用的直流电动机进行叙述。

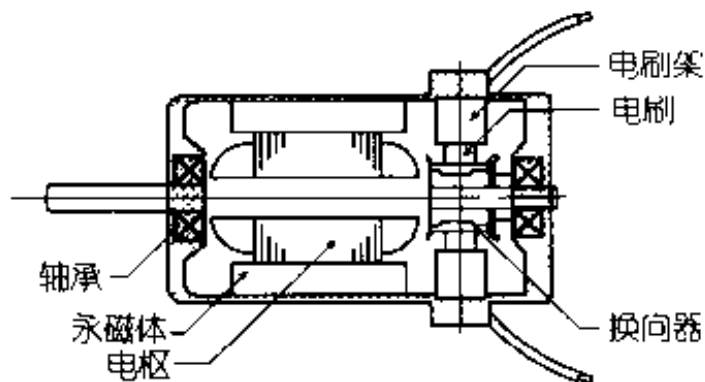
在相扑机器人中,使用的电动机大部分是直流电动机,其中一部分使用步进电动机或超声波电动机。步进电动机在3相、4相等线圈中,通过依次流过的电流使其旋转,所以没有电刷等。另外,由于可以用开环控制,所以对初学者制作机器人适宜。但是,由于步进电动机的转矩小或失步等,对相扑机器人不适合,所以这里不必阐述。

8.1.1 电动机的工作原理

了解电动机的工作原理,就可以研究在各种各样的用途中怎么做,能确定使用简单的电动机。

如图8.1所示,直流伺服电动机由固定的永磁体和铁芯上缠绕线圈的电枢构成。电枢由连接端部的线圈、接触电刷部分的换向器和轴组成。由于从外部流入电流,所以电刷和换向器相互接触。

在由永磁体产生的磁场中,线圈先固化好。这个磁场的强度和流过线圈电流的强度、线圈匝数与电动机产生的转矩成正比例。当电流流过线圈(根据弗莱明左手定则)时,转子开始运动,电流流过下一个线圈,转子继续运动,通过上述依次运动,电动机就可以连续旋转起来。



• 图 8.1 电动机的结构

在电气方面如上所述,用弗莱明左手定则判断,首先产生转矩。电动机开始旋转时,取决于下次磁场中线圈运动。为此,这个线圈根据弗莱明的右手定则应该产生电压。我们将这个电压称为反电动势,将这个反电动势与外施的端电压相对应,在线圈中感应电流。根据这个感应电流确定电动机中产生的转矩。因而,这个反电动势大时,电动机最高转速变大¹⁾。

1) 原书中有误。因为直流伺服电动机中的 $E_b = C_e \phi n$, 所以当直流伺服电机完成后, $C_e \phi$ 为常数。则 $n \propto E_b$ 变化。即反电动势 E_b 越大,则电动机的转速 n 就越大。——译者注

列举数式如下,这里如果能够理解,就如掌握电动机原理一样。即使不完全掌握电动机原理,通过试运行找出错误,以实现电动机的控制。

1. 电动机产生力*F*根据弗莱明定则列式如下:

$$F(t) = 2\pi rNBi(t) = KT i(t)$$

其中,*N*为线圈匝数;*r*为线圈半径;*i*(*t*)为电流;*KT*=2π*rNB* 为转矩常数、推力常数。

2. 电气方程式,根据基尔霍夫定律下式成立:

$$L \frac{di(t)}{dt} + Ri(t) = v_a(t) - e_i(t)$$

其中;*L*、*R*分别为线圈的电感系数和电阻;*v_a*(*t*)为线圈上施加电压;*e_i*(*t*)为反电动势,*e_i*(*t*)=2π*rNBv*(*t*)=*K_e**v*(*t*)。其中,*v*(*t*)为线圈移动速度,*K_e*=2π*rNB*,为反电动势常数。

将电动机两端短路,用手旋转电动机时,可以体会到这个反电动势大小。即磁场强的磁体或线圈匝数多的电动机需要制动时反转很困难。这个制动作用就是来自反电动势。

为了提高电动机的最高转速,在外施端电压不变时,只要减少匝数即可,这是无线方向遥控用的高输出功率电动机。但是,即使短路制动也很难起作

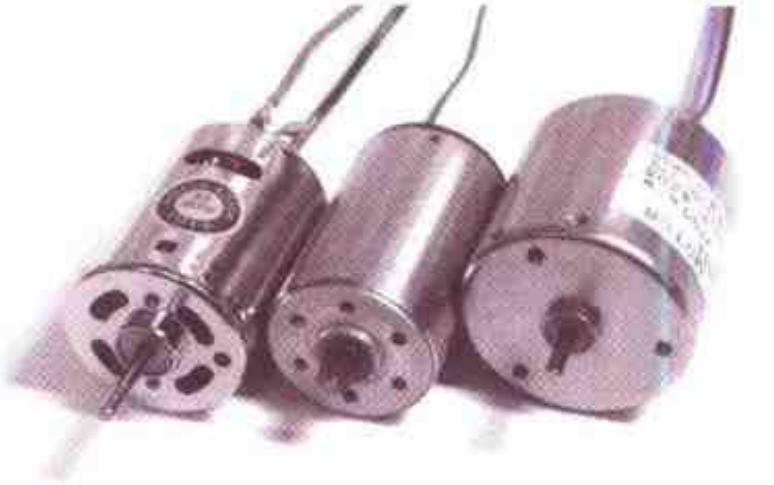
用。对控制用伺服电动机匝数变大,对制动效果旋转起来后可以控制。

在实际的电动机控制中,考虑电动机的惯性矩等,也需要考虑机械系统的运动,而这里省略。

8.1.2 直流(DC)电动机的种类

在机器人中,下面试比较它们的性能。经常使用的电动机有日本伺服DME34BA 和其尺寸近似的 MABUCHI 电动机及 Escap 高级电动机。下面试比较它们的性能。

表 8.1 表示几种电动机的特性。这里很大的差别是电压、输出功率和转速等。特别是 MABUCHI 电动机输出功率大、转速高。这正如上节的说明,就是由于匝数和线径的差别而产生的。



• 照片 8.1 电动机的比较(从左开始 MABUCHI 电动机、Escap、日本伺服电动机)

表 8.1 电动机特性比较

制造厂名	型式	额定值					空载		停止 转矩/ (N·m)	重量/ g
		输出 功率/ W	电压/ V	转矩/ (N·m)	电流/ A	转速/ (r/min)	电流/ A	转速/ (r/min)		
日本伺服电动机	DME34BA	4.5	12	0.0118	0.78	3700	0.1	5000	0.0451	100
日本 Voltescap	26L28-216P	5.7	6	0.017	1.52	3500	0.032	4900	0.0293	107
MABUCHI 电动机	RS-380PH	18	6	0.0113	3	10000	0.48	11700	0.0853	80

8.1.3 转 子

在照片 8.2、照片 8.3 中表示带有线圈的转子部分,其特性在表 8.2 中表示,而它们各自的直流电阻有很大的差异。另外,线径也不同。Escap 电动机是高级产品,所以没有卸开,其内部结构如图 8.2 所示,是无铁芯电动机,回转部分惯性小、响应快,作为控制用电动机使用。

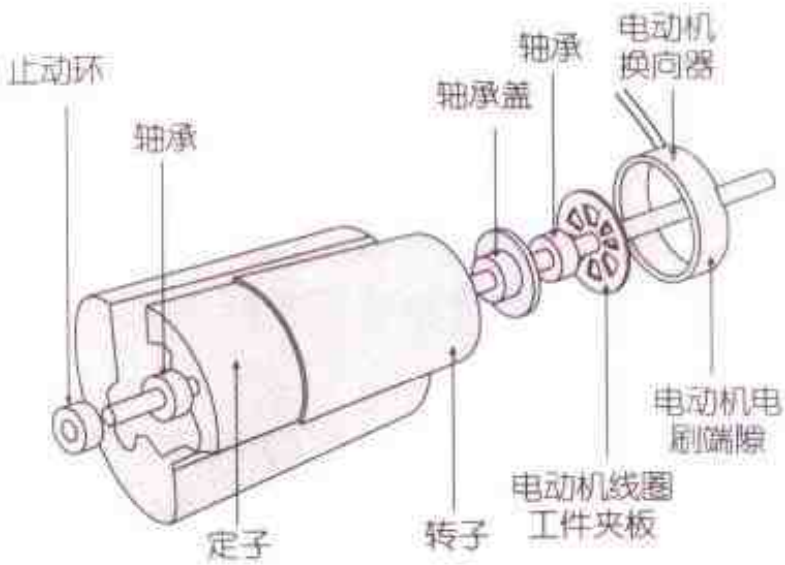
这里电阻值是从外部测定的。



• 照片 8.2 日本伺服电动机的转子



• 照片 8.3 MABUCHI 电动机转子



• 图 8.2 Escap 无铁芯电动机(从 API 日本 Voitescap 的商品目录上摘录)

表 8.2 转子线圈的比较

制造厂名	直流电阻	线径	匝数
MABUCHI	0.7Ω	0.4mm	45
Escap	2.8Ω	—	—
日本伺服电动机	5.4Ω	0.2mm	200

8.1.4 永磁体

照片 8.4、照片 8.5 中表示电动机中用的永磁体。有关永磁体的特性不清楚,可以认为磁力大体相同。在无线遥控车爱好者的世界,这种磁力提高的装置等好像也有出售。使用贵重稀土类永磁体可以作成高性能永磁体。这样,磁密大幅度提高,在一部分精密电动机中已经使用(稀土永磁体)。

8.1.5 电 刷

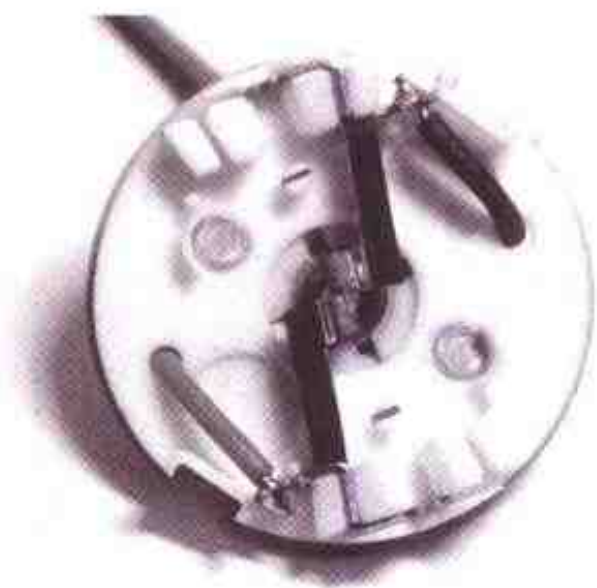
照片 8.6、照片 8.7、照片 8.8 表示各自电动机的电刷,电刷对应其流过的电流、大小不同。特别是在 MABUCHI 电动机中,为了降低电刷的接触电阻,而采用大接触面积的电刷。



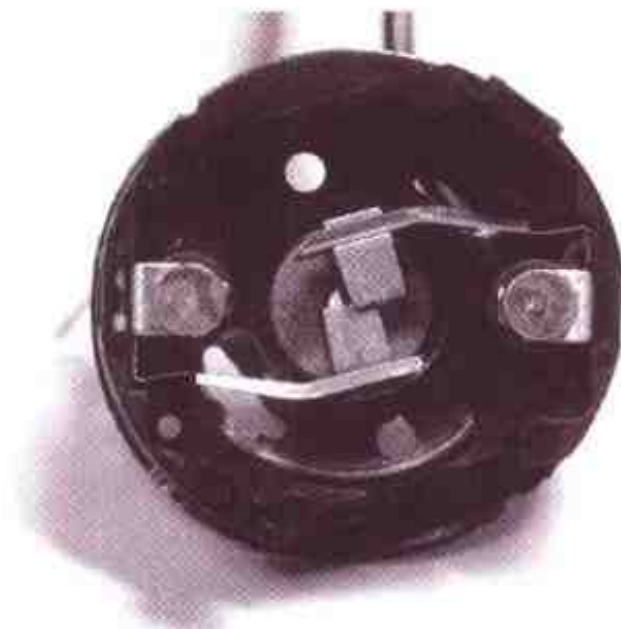
• 照片 8.4 日本伺服会社的电动机中的永磁体



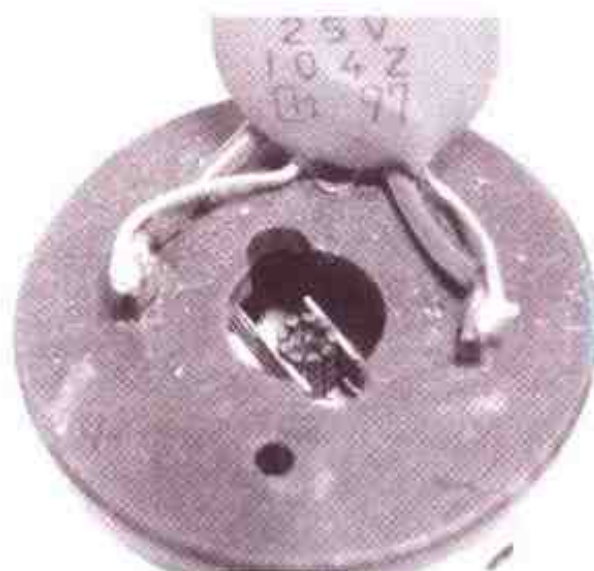
• 照片 8.5 MABUCHI 电动机永磁体



• 照片 8.6 日本伺服会社电动机的电刷



• 照片 8.7 MABUCHI 电动机的电刷



• 照片 8.8 Escap 社电动机电刷

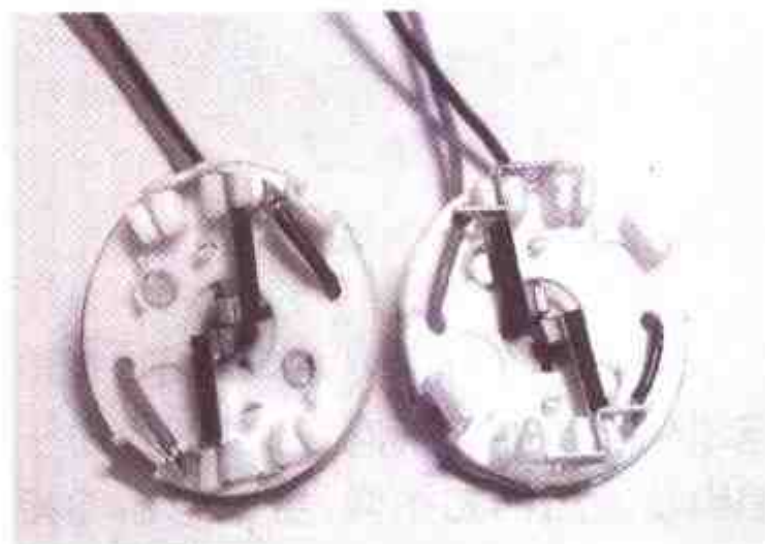
在电刷中有旋转方向问题,对将电动机向右旋转或向左旋转时,由于给电刷施加的力不同,所以电刷的接触面积不同,由此,流出的电流不同,这就形成转矩的差别。在 MABUCHI 电动机中,规定了旋转方向。

在相扑机器人中,将电动机安装在车轮的右边或左边,前进的时候电动机左、右旋转。因而,由于接触电阻不同,所以往往转速也不同,这正是机器人可以左右移动的原因。只要为了使其转速相同,改变电刷的方向即可。

日本伺服会社电动机,电刷部分由于铆接在电动机的机身上,所以必需注意经常地拆卸。对于这个电动机,由于有相反安装的槽沟,所以如照片 8.9 所示,用扁嘴钳子将电刷取出,然后再如照片 8.10 那样,将电刷反方向插入即可。



• 照片 8.9 拆卸电刷



• 照片 8.10 变更电刷的旋转方向

8.1.6 高转矩电动机

为了提高电动机的转矩,从电动机的工作原理出发,可以考虑如下因素:

① 使线圈流过大电流(提高电压,将线径变粗);

② 使定子永磁体的磁场变强(更换磁场强度大的永磁体);

③ 增加线圈的匝数;

④ 降低电刷的接触电阻(安装大面积电刷);

⑤ 提高转速、增加传动比。

电动机制造厂的网址

MABUCHI 电动机株式会社

<http://www.mabuchi-motor.co.jp/mot/index.html>

狭山精密工业株式会社

<http://www.sayama-pre.co.jp/seihin.htm>

API(Voltescap)

<http://www.apimotion.com/API日本Voltescap>

TEL:03-3241-0201

Maxon 日本株式会社

<http://www.maxonjapan.co.jp/>

光进电气工业株式会社

(微型电动机的代理店)
<http://www.koshindenki.com/>

微型电动机
<http://www.minimotor.ch/>

并木精密宝石株式会社

<http://www.namiki.co.jp/motor-e.htm>

日本伺服

<http://www.japanservo.com/>

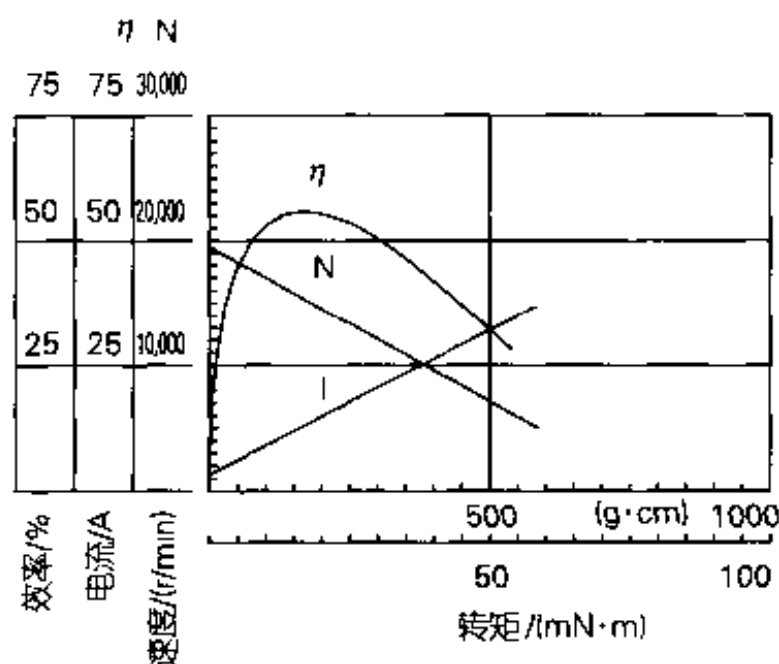
田宫模型

<http://www.tamiya.com/japan/robocon/robocon.htm>

8.2 电动机驱动器

8.2.1 电动机的特性

将既能控制流过电动机电流大小或流动方向,又能驱动电动机的电路称为电动机驱动电路。这个电动机驱动电路,必需对照电动机的特性选定,特别重要的是电动机的电流特性。



• 图 8.3 RS-540 MABUCHI 电动机的特性

图 8.3 是 MABUCHI 会社制作的 RS-540 的特性。从图中可知,随着负载的上升电流增加,转速下降。并且电动机的电流对应负载变化,负载越大,电流越大。

在相扑机器人中,和对手相互正面接触,往往看到电动机停止等的状态。对停止状态电动机的直流电阻部分,即由线圈的电阻和电刷的接触电

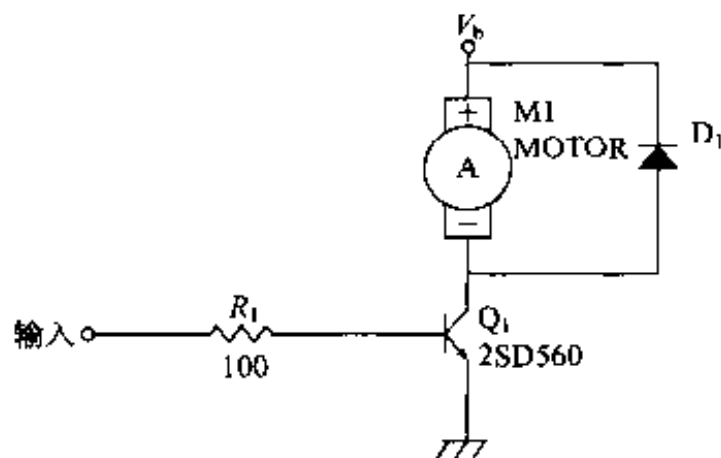
阻、连接电缆及电池的内阻决定流过电流。对 RS-540,在电动机停止状态时,电流应该超过 20A。即使这种状态,为了不损坏电动机驱动器,必需精心地设计。在选定电动机驱动器时,最重要的是这个电流。当电动机电压上升时,随着电流继续不断地增加,转矩上升。因此,就希望电压继续不断地上升,通常,在机器人中使用电动机驱动器要有一定余量(这里指使用晶体管的电压值),所以多数情况设定为 50V 左右的电压。往往初学者容易把电压考虑为额定电压,且忽视电流,这样一来,完全有可能损坏电动机驱动器。

另外,不能只看电动机额定电压、额定电流选定电动机驱动器。额定值是经常使用最高的效率,其最好是考虑所谓的推荐值。对机器人必需考虑好各种各样的电动机的使用状态。在电动机驱动器的选定方面,为了尽可能发挥电动机的功率,必需首先考虑电动机的堵转电流(不用说,还包括电池的电流)。

8.2.2 旋转电动机的方法

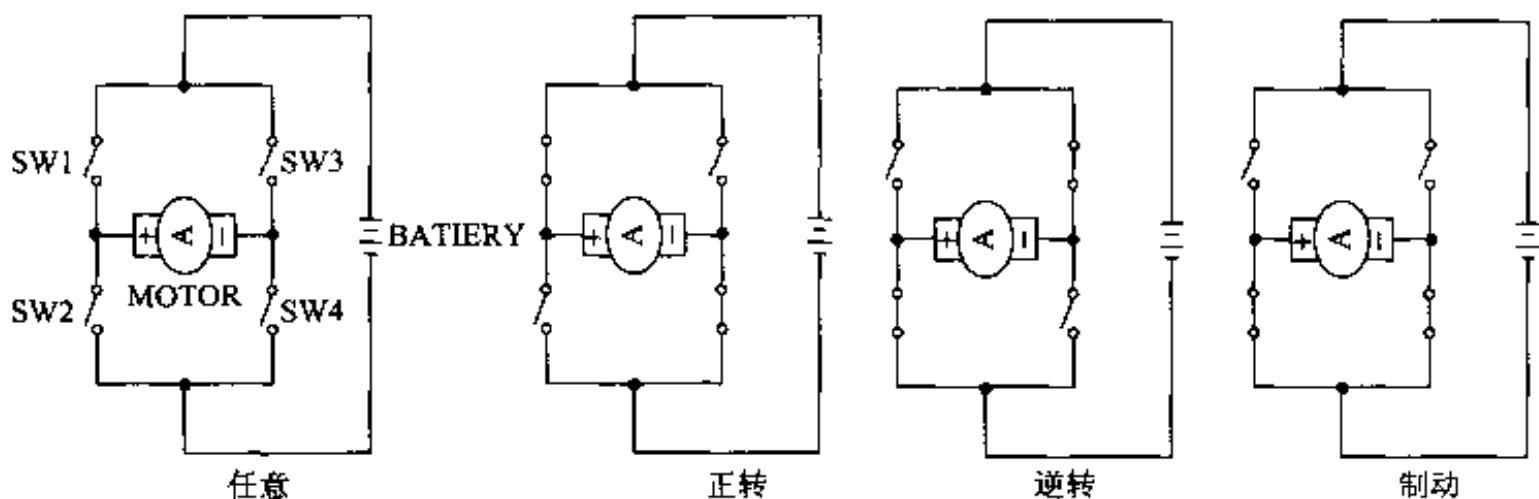
为了旋转电动机,只要连接电动机的电池即可,为了控制它,使用由晶体管组成的开关。这个电路如图 8.4 所示,由于输入侧为高电平,所以电动

机旋转。但是,这里只能按同一方向旋转,最好仅使用在前进机器人上,然而为了使电动机能正转、反转,通常构成桥式电路。构成桥式电路时,不仅电动机能正转、反转,而且还可以制动。



• 图 8.4 电动机的驱动电路

图 8.5 表示了电动机和开关的连接状态。对打开状态,由于什么也没连接,所以电动机不旋转。但是,当有外力作用时,电动机即使被旋转,也不可能制动,只是自由地旋转。对正旋转,电流从电源正极由电动机左边开关 SW1 流向右边开关 SW4,返回电源的负极。对反转旋转,电流从电源的正极由电动机右边开关 SW3 流向左边开关 SW2,返回电源负极。对于制动状态,电动机线圈两端由开关 SW2 和 SW4 短路,所以通过反电动势形成制动状态。这种用晶体管构成的电路叫做电动机驱动电路。



• 图 8.5 驱动电动机桥式电路的工作原理

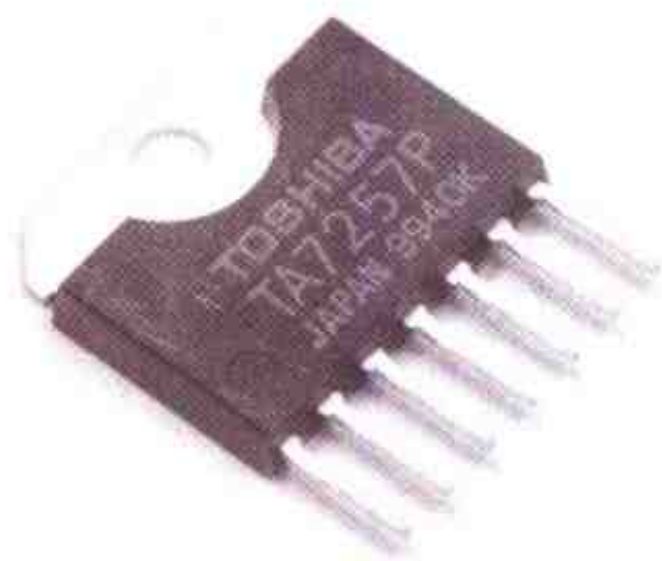
8.2.3 驱动电动机 IC 模块

由于用晶体管构成的桥式电路为一个芯片 IC,连接简单、使用容易,所以在很多机器人中使用。

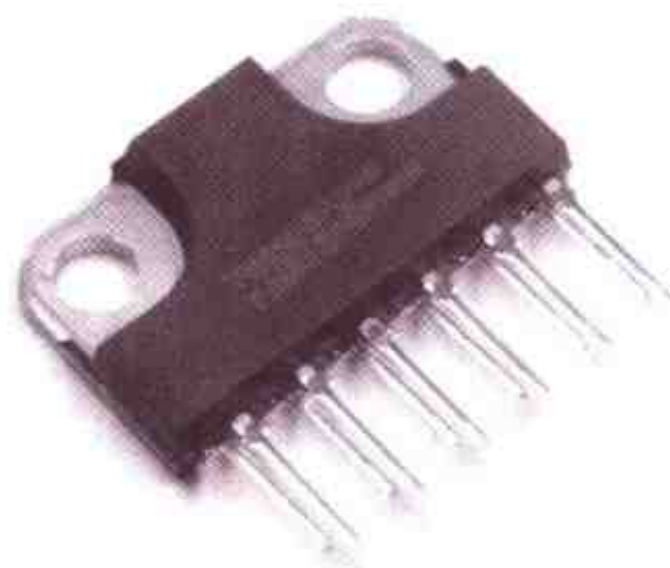
照片 8.11 是东芝 TA7257、平均电流为 1.5A 的电动机驱动器。如图

8.6 在一个芯片上组装 4 个晶体管,构成不同的晶体管桥,容易制作。对直线追踪机器人,即使这样大小的电流也可以使用。

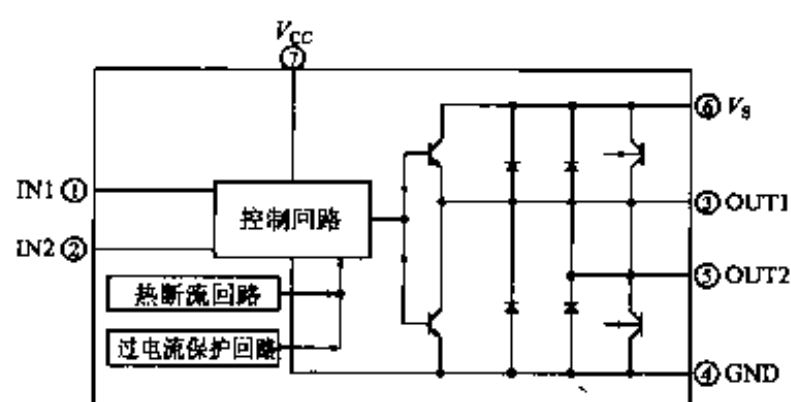
表 8.3 表示电动机的控制方法,而用 2 位(1、0)的输入信号,就能控制正转、反转、制动和停止状态。



• 照片 8.11 东芝的 TA7257P



• 照片 8.12 东芝 TA8429H 驱动器



• 图 8.6 内部电路

表 8.3 控制方法

IN1	IN2	OUT1	OUT2	状态
1	1	L	L	制动
0	1	L	H	正/反转
1	0	H	L	反/正转
0	0	高阻抗		停止

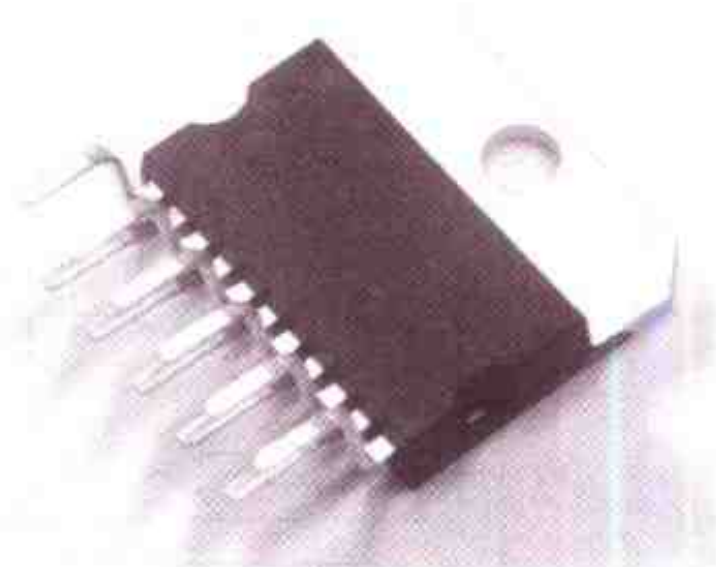
照片 8.12 是东芝 TA8429H 驱动器。表 8.4 中表示了它的电气特性，其为采用平均电流到 4.5A 的电动机驱动器 IC。在机器人领域，被广泛使用。

照片 8.13 的 IC 是 L6203 驱动器。这个 IC 是 ST 微型电子设备会社制作。

表 8.4 TA8429H 的电气性能

最大规格($T_a = 25^\circ\text{C}$)

项目	符号	额定值
电源电压/V	V_{CC}, V_S	30
输入电压/V	V_{IN}	$-0.3 \sim V_{CC}$
输出电流/A	AVE	$I_O(\text{AVE})$
	APEAK	$I_O(\text{PEAK})$
消耗功率/W	P_D	2.25 ²⁾
		21.6 ³⁾
工作温度/ $^\circ\text{C}$	T_{OPR}	$-30 \sim 85$
保存温度/ $^\circ\text{C}$	T_{STG}	$-55 \sim 150$

1): $t = 100\text{ms}$; 2): 没有散热片; 3): $t_c = 85^\circ\text{C}$ 

• 照片 8.13 L6203 驱动器

从 ST 微型电子设备会社网页可以获得数据图表

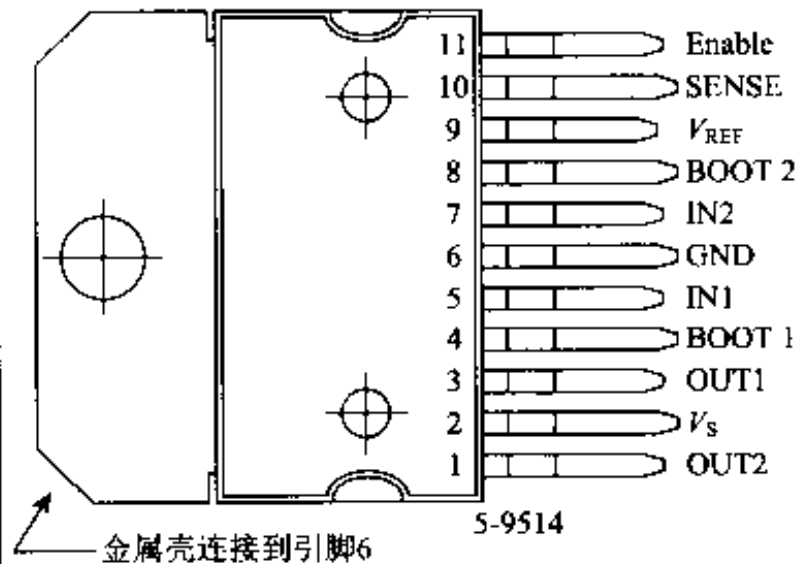
<http://www.st-japan.co.jp/>

TA8429H 是双极晶体管型电动机驱动器,对于这个驱动器的 IC 模块由 MOS-FET 构成, MOS-FET 具有优越的高频特性。

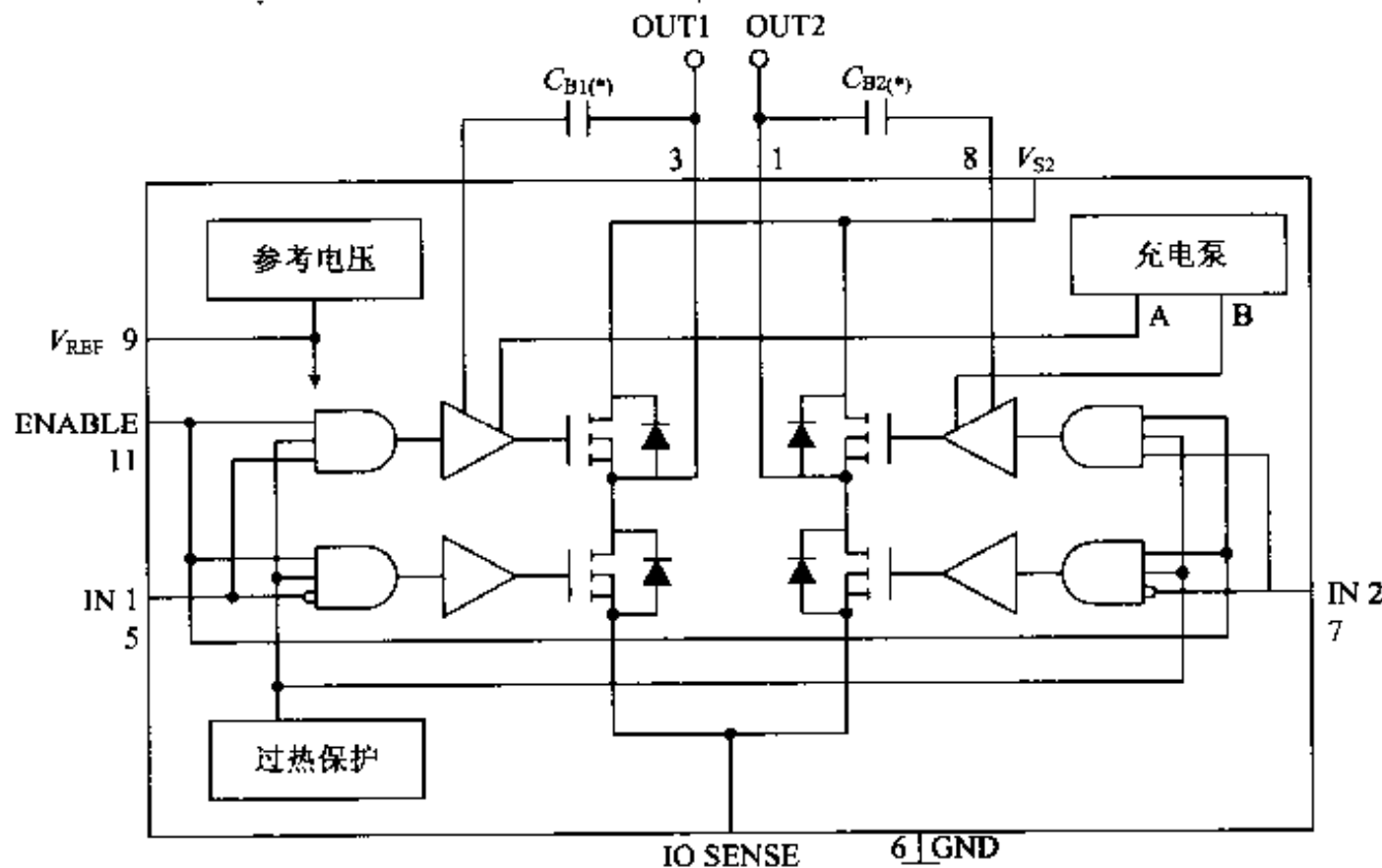
这个电动机驱动器,导通(ON)电阻控制在 0.3Ω ,且发热小,使用简单。供电电压可以使用到 48V,驱动电动机的平均电流可以达到 4A。另外,峰值电流可以谐调到 5A。

有关这个 IC 引脚进行一下说明。图 8.7 是 L6203 的外形,在引脚上标有名称。表示其内部电路如图 8.8 的方框图。OUT1、2 端连接电动机,电源的正、负级分别连接 V_S 、GND。为了控制这时电动机正转、反转和停止,

信号为 Enable 引脚和 IN1、IN2 引脚。设 Enable 引脚电压为 5V 时,可以控制电动机旋转。相反,设其电压为 0V 时,电动机停止。设 Enable 引脚电压为 5V 状态,IN1 引脚电压为 5V,IN2 引脚电压为 0V 时,电动机正转。相反,设 IN1 引脚电压为 0V,IN2 引脚电压为 5V 时,电动机反转。



• 图 8.7 L6203 IC 的外形

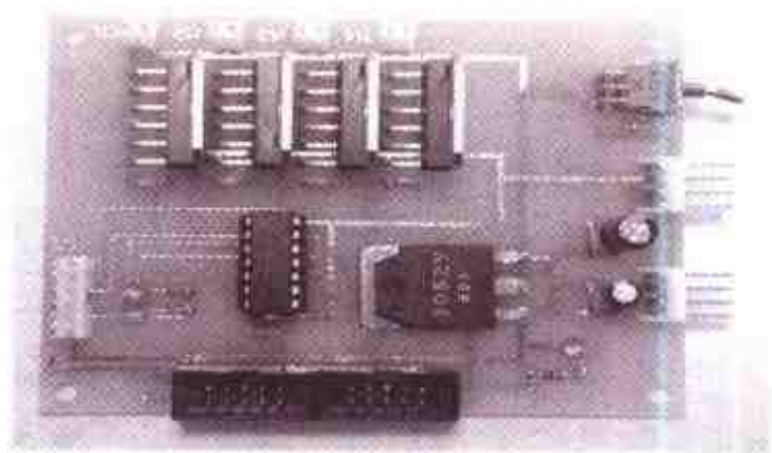


• 图 8.8 L6203 IC 方框图

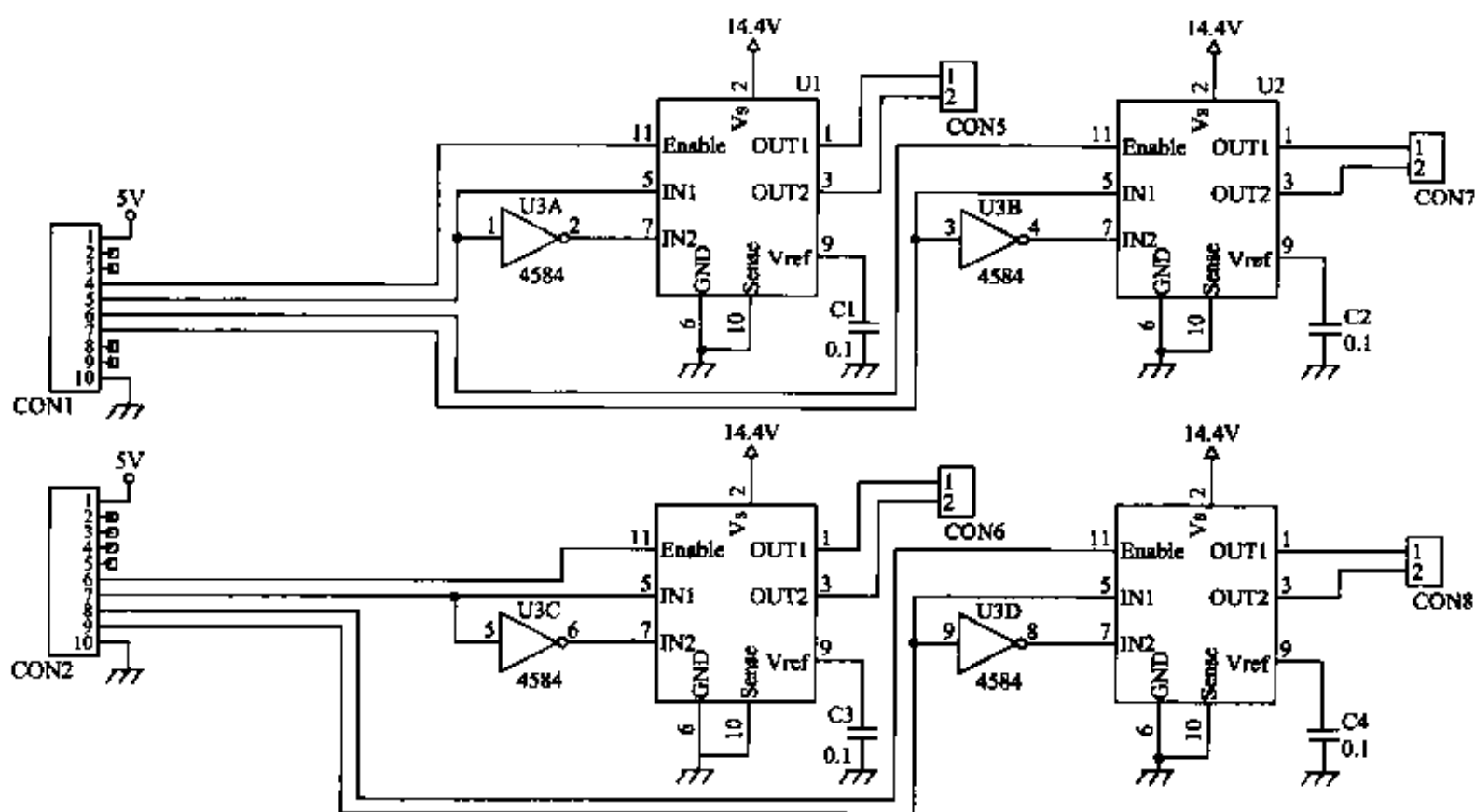
8.2.4 根据 L6203 制作驱动电动机插件板

使用 L6203 制作驱动电动机插件板,图 8.9 为驱动电动机插件板的电路图。在这个电路中,对于 L6203 的 IN1 引脚,在 IN2 引脚上输入 IN1 引脚的反转信号。由此,用一个控制信号,就可以控制正转,反转。这里就是输入 PWM 信号,从正转到反转进行平滑的控制。用高频变换正、反转,通过变化其正转时间和反转时间的比率,可以微妙地一边进行正转,一边慢

慢地进行反转和制动。照片 8.14 是制作完的驱动电动机插件板,内装 4 个 L6203IC 模块。这个电动机驱动器好像可以使用在各种各样的用途当中。



• 照片 8.14 制作完的驱动电动机插件板



• 图 8.9 驱动电动机插件板电路图

8.2.5 使用双极型晶体管驱动电动机电路

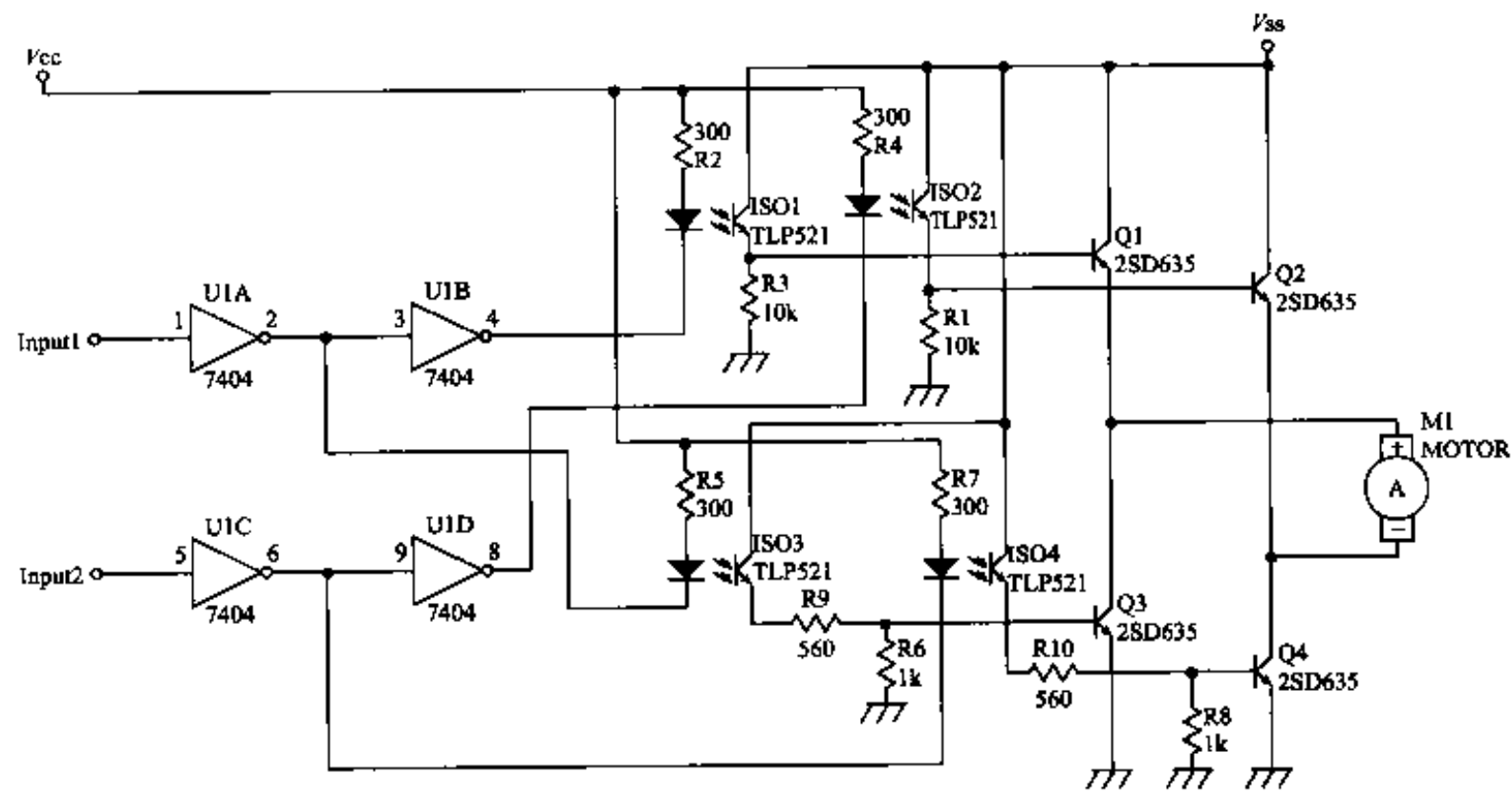
图 8.10 是使用双极晶体管的驱动电动机(M1)电路图,晶体管使用 2SD635。这个晶体管最高电压为

60V,最大电流为 7A。在图 8.11 中表示出其外形尺寸。

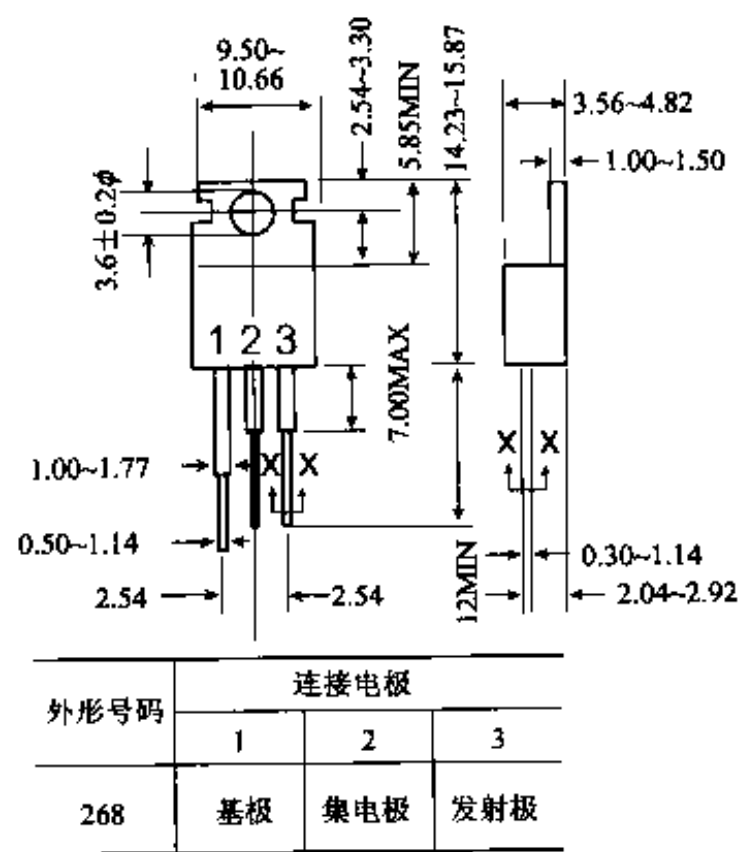
如果进一步改变这个晶体管直到它的最大电流值,并且可以使用这个大电流,制作驱动电动机电路。这里

使用了现有的 2SD635 晶体管,而在这个晶体管规格表内选择几个电流值大

的晶体管,到秋叶原元件商店询问一下就可以知道,只要使用它即可。



• 图 8.10 使用双极型晶体管的驱动电动机电路



• 图 8.11 2SD635 的外形尺寸

在这个驱动电动机电路中,使用光电耦合器(参照照片 8.15 所示),将 CPU 电源和驱动电动机用的电池电源

与电分开。为此,由于驱动电动机时,电池电压的下降,可以解决 CPU 的清 0 或由驱动电动机时产生噪声问题。最近 CPU 抗噪声性变强,为了安全,驱动大电流电动机时,使用它即可。



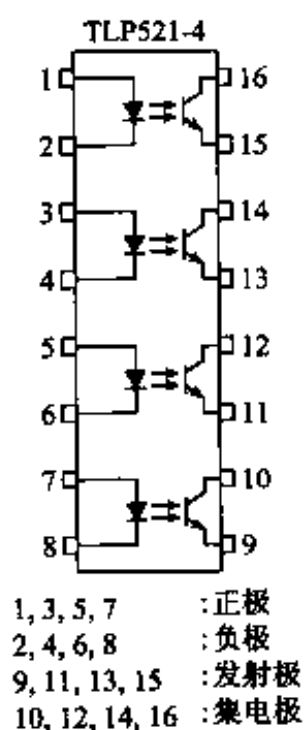
• 照片 8.15 光电耦合器

正如光电耦合器的名称那样,通过光传输信号,所以即使不通电信号,将驱动电动机用的晶体管通过计算机也可以控制。光电耦合器如照片 8.15

所示,和 TTL 封在一起,引脚配置如图 8.12 所示,为 4 组光电晶体管和光电二极管封装成一个模块。这里使用的 TLP521-4 的详细规格表从东芝的网页上可以下载。

东芝网页

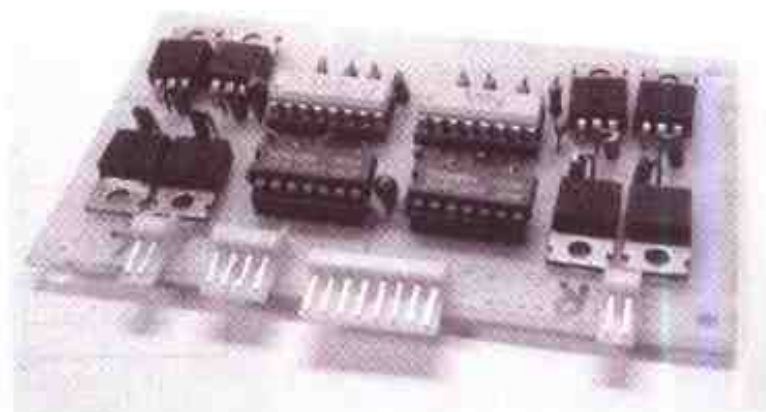
<http://www.semicon.toshiba.co.jp/noseek/jp/td/14frame.htm>



• 图 8.12 光电耦合器引脚配置

在这个驱动电动机电路中,由 CPU 侧驱动光电二极管 ON-OFF。通过这个光控制连接电动机侧电源上的光电晶体管的开、关。为此,应该使功率晶体管 2SD635ON-OFF。从 CPU 侧为了不能使晶体管 TR1、TR2 或者 TR3、TR4 同时 ON,通过 TTL-IC 组成逻辑模块。这里用 2 个引脚 ON-OFF 进行电动机的控制。

将这个电路内装 2 个回路,作成的插件板如照片 8.16 所示。



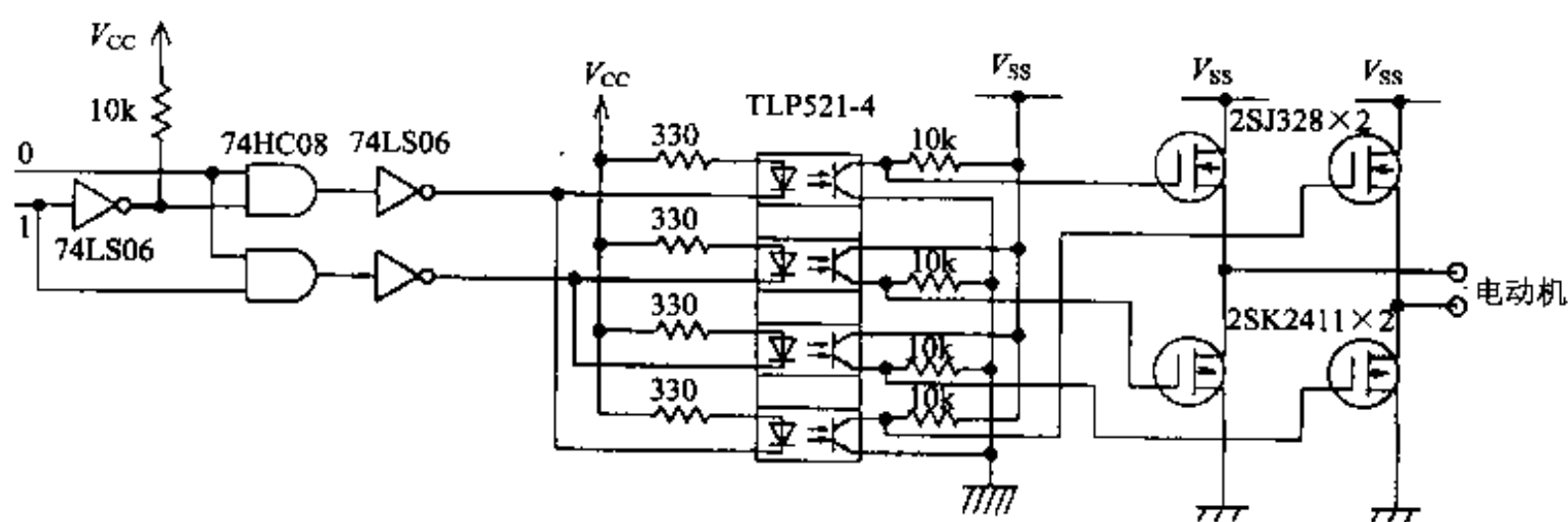
• 照片 8.16 制作完的 2 回路驱动电动机电路

8.2.6 使用 FET(功率场效应晶体管)驱动电动机电路

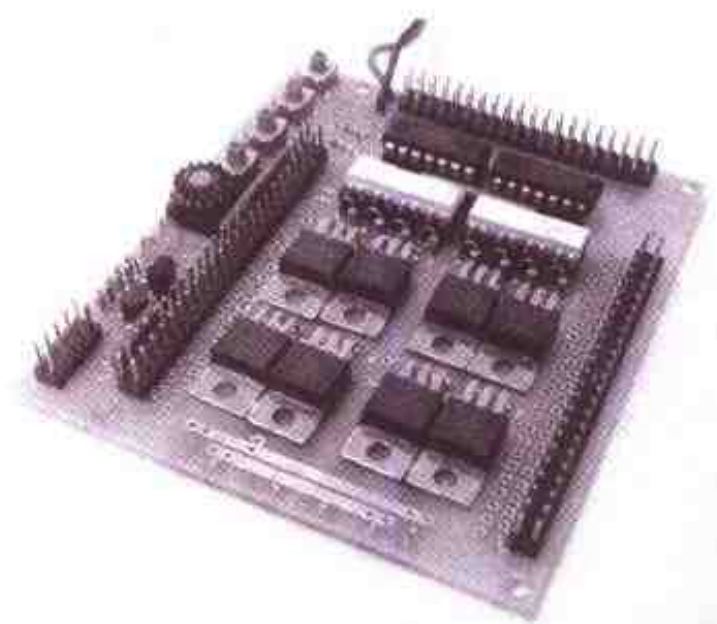
图 8.13 中表示使用 FET 的驱动电动机电路的实例。FET 使用的是 2SJ328 和 2SK2411 功率场效应晶体管。2SJ238 最大电压、最大电流分别为 -60V、20A, 2SK2411 为 N 沟道的 FET(功率场效应晶体管),其最高电压、最大电流分别为 60V、30A。这些详细的规格从下面的 NEC 网页上可以得到。

<http://www.ic.nec.co.jp/discrete/index.html>

照片 8.17 是使用 FET 的内装 2 个驱动电动机回路的插件板。而且为了高输出功率化,只要将 FET 功率场效应晶体管并联连接即可。如果分别连接栅极、漏极和源极,使流过漏极电流增加 2 倍,能连接 3 个 FET 变成 3 倍的漏极电流。



• 图 8.13 使用 FET 驱动电动机电路



• 照片 8.17 内装 2 个 FET 驱动电动机回路的插件板

FET 并联连接的功率场效应晶体管电流分别均衡,流畅地流过功率场效应晶体管。由于这样的理由,用它可以构成高输出功率的 FET 功率放大器。

电动机驱动器制造厂的网址
 东芝电动机驱动器
<http://www.semicon.toshiba.co.jp/noseek/jp/td/06frame.htm>
 ST 微型电子设备会社
<http://www.st-japan.co.jp/>

对双极晶体管并联连接时,电流有完全集中在一个晶体管上的特性,而对

8.3 电动机的控制

1. 使用集成定时单元 (ITU) 的 PWM 信号控制电动机

使用 ITU 产生 PWM 信号,进行电动机的速度控制。驱动电动机电路使用图 8.9 所示的电路。在驱动电动机 IC 上使用 L6203 模块,设 Enable 引脚

为高电平时,在另一个引脚输入 PWM 的信号,就可以驱动电动机。在驱动电动机 IC 的 IN1 和 IN2 中,由于通过逆变器,所以应该输入相反的信号。

即从表 8.5 中的 L6203 真值表中可知,IN1 引脚为 5V 时正转,为 0V 时

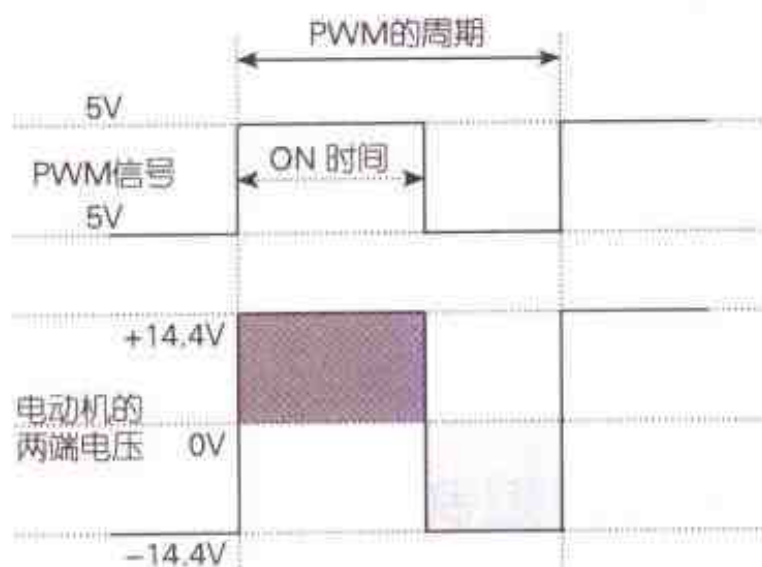
应该变成反转。

表 8.5 L6203 的真值表

IN1	IN2	电动机的旋转方向
5V	0V	正 转
0V	5V	反 转

2. PWM

所谓 PWM 就是脉宽调制, 这里如图 8.14 所示。在周一周期改变输出 ON-OFF 信号的 ON 和 OFF 时间的比率, 改变流过电动机的平均电流, 就是希望控制电动机的转速。在驱动电动机插件板上施加 PWM 信号时, PWM 信号 ON 期间串联连接 2 个无线方向遥控用的 7.2V 蓄电池, 所以电动机两端电压这时变为 14.4V。另外, OFF 期间电压变为 -14.4V, 在电动机中流过正、负电流。即通过图 8.14 中的■部分和□部分差产生的平均电流, 使电动机旋转。

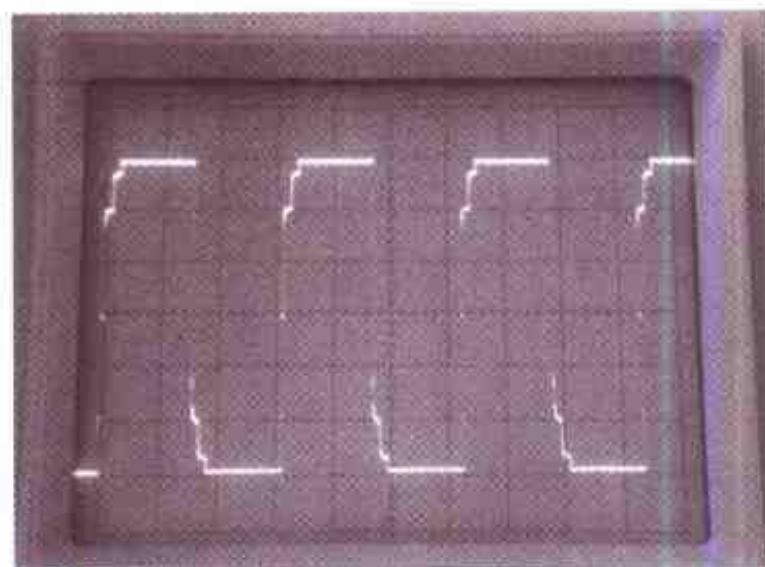


• 图 8.14 通过 PWM 信号产生电动机两端电压

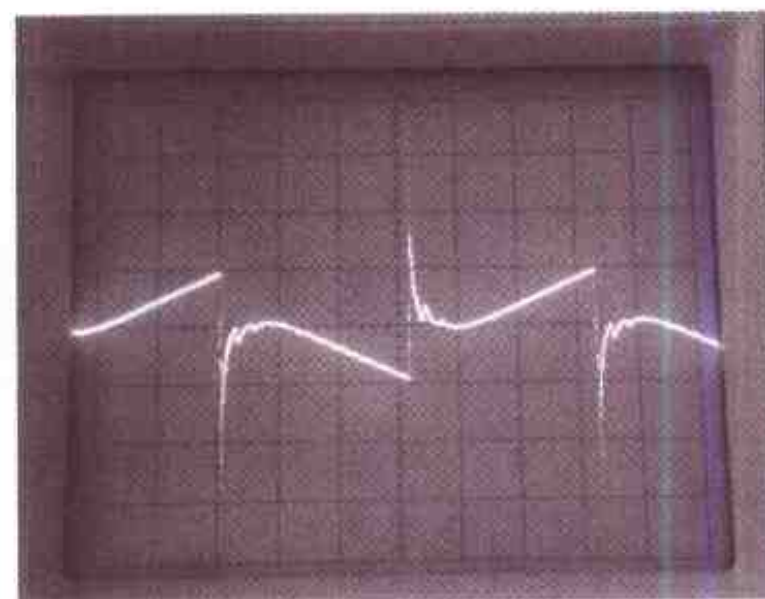
图 8.14 中的■和□部分相等时,

即 ON 时间和 OFF 时间相同时, 电动机停转。使 ON 时间变大下去时, 电动机正转, 逐渐地增加电动机的速度。ON 时间减少时, 电动机反转, 在 ON 时间接近 0 时, 电动机开始反转, 并且反向增加速度。这样, 就是通过改变 PWM 的 ON 时间, 使电动机从正转到反转, 仅仅用一个输出信号就可以进行控制, 于是, 就可以实现机器人平滑地运动。

我想有人担心, 尽管电动机处于停止状态, 也要流过很大的电流, 要消耗电池的很多能量。如照片 8.18、照片 8.19 所示的 ON 时间和 OFF 时间



• 照片 8.18 停止时的电压波形



• 照片 8.19 停止时的电流波形

相等时的电压和电流波形。由于电动机线圈中事先产生电感,它具有即使施加电压,电流也不能马上流过的特性。为此,PWM 的周期十分迅速时,消耗电流很少。即使这么说,与电动机完全 OFF 情况相比,消耗能量也稍

微变大。

程序

将驱动电动机插件板连接在微型计算机的端口 PA、PB 上,试使用 PWM 信号控制电动机。

简 语

1) 噪声对策

驱动电动机时,噪声是一个突出问题。

由于电动机反复地进行 ON 和 OFF,往往产生尖峰状的电压。这时,CPU 有误动作的情况。在电动机的两极间加入 $0.1\mu\text{F}$ 左右的电容器,往往可以解决 CPU 的清 0 等问题。还有,流过大电流的时候,将 CPU 和电动机的电源分开,将信号线采取光电耦合器的方法即可。

2) 原因不详的 CPU 清 0 对策

是什么原因不清楚,而时常 CPU 遭遇清 0 的情况时,有用程序避开的方法。在相扑机器人中,电源电压降低时,CPU 遭遇清 0,这件事在机器人竞赛大会召开前几天才了解到。改造硬件的时间没有了,所以 CPU 即使清 0,也要有使机器人继续运动的程序。对于通常的程序,遭遇到清 0 时,变成保持起动 SW 的状态,机器人变成停止状态。这时加入检测 Dip SW 程序,对战中即使 CPU 遭遇清 0,也转移到对战程序上。如果机器人停止运动,无论对手多么软弱都不能战胜,只有非常小的概率只要出手即可以取得胜利。在相扑机器人大会上,可以想像仍会出现在机器人调试中出现的故障。

8.4 机器人臂结构和控制

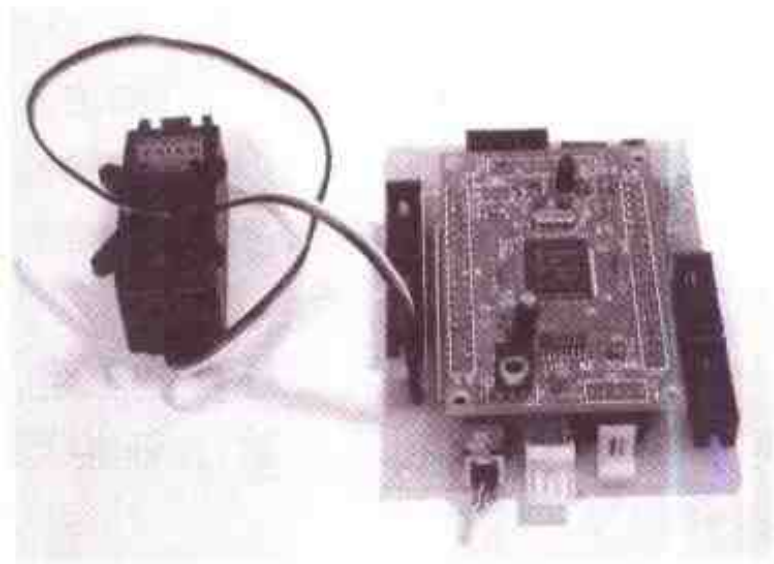
8.4.1 无线遥控用伺服电动机的使用方法

使用无线遥控用伺服电动机,可

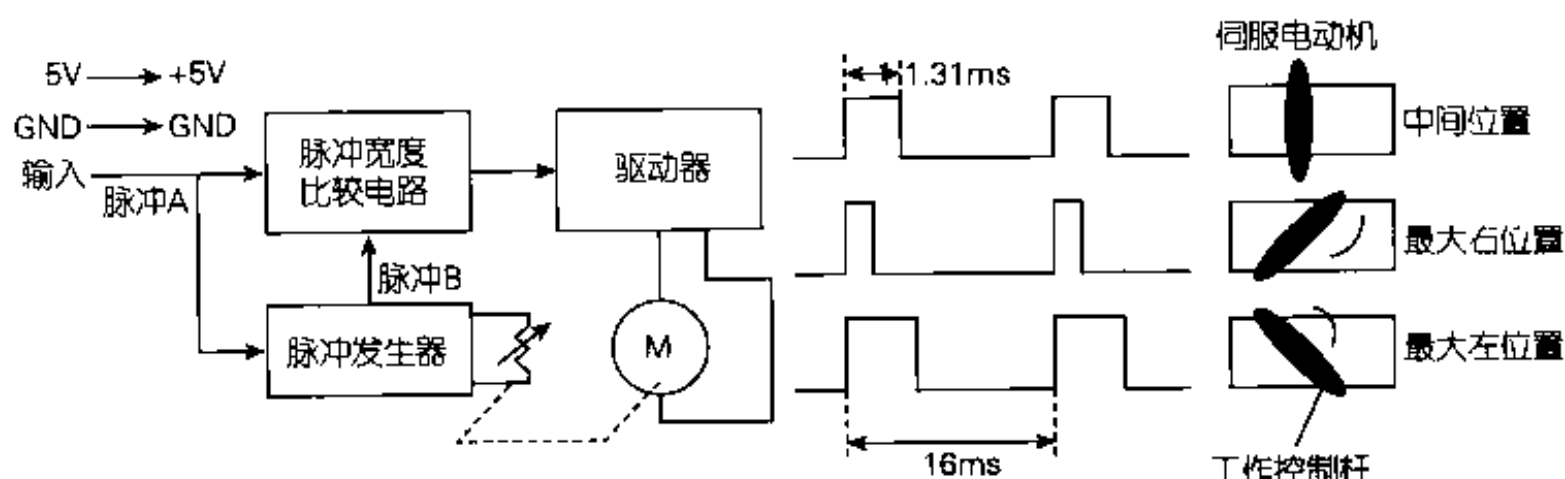
以控制直线追踪机器人的转向,这里例举无线遥控用伺服电动机控制实例。

图 8.15 中表示无线遥控用驱动

伺服电动机的工作原理图。图中左边是驱动伺服电动机的方框图,右边是控制方法。在伺服电动机上如果输入 PWM 信号,对应其脉冲宽度伺服电动机的角度变化。这个 PWM 的周期为 16ms,伺服电动机中间位置在脉冲宽 1.31ms 的地方,将这个脉冲宽度只要通过微型计算机控制即可。伺服电动机的信号线中红引线为 +5V,黑引线接地,白引线为信号线。



• 照片 8.20 伺服电动机控制插件板



• 图 8.15 无线遥控用伺服电动机控制

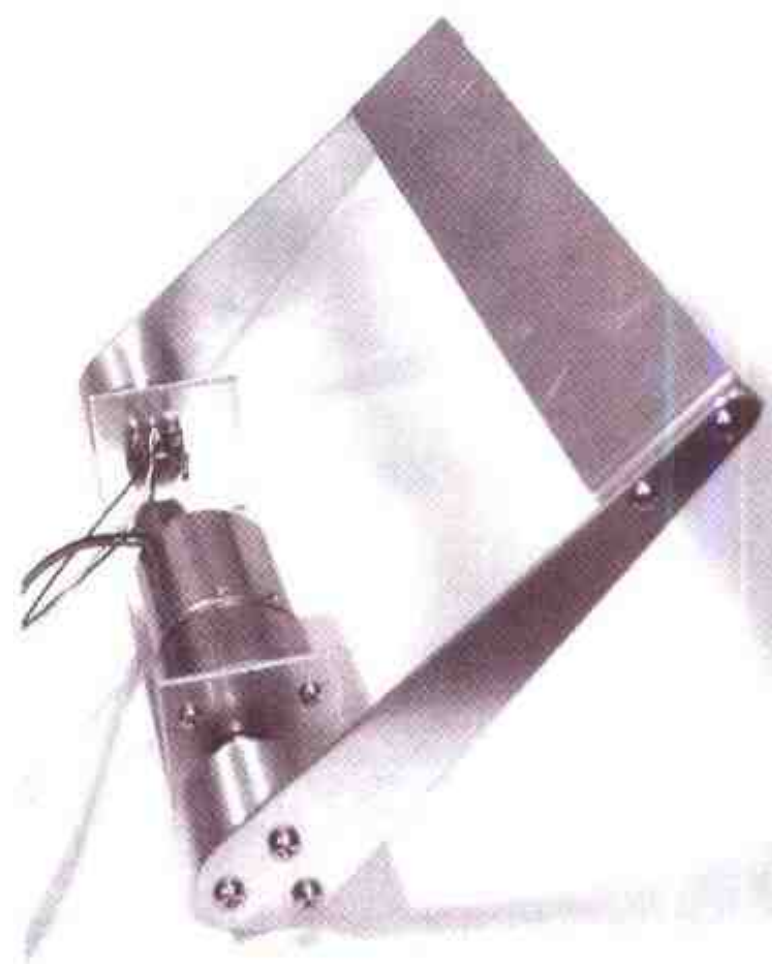
程序

根据数字开关值,试编制伺服电动机角度变化的程序。

8.4.2 臂的结构

臂结构如照片 8.21 所示,它是将电动机和电位器(可变电阻)装配在一起的简单结构。通过旋转电动机、电位器旋转,变化电位器的输出。看电位器输出,将电动机用 PWM 信号进行确定位置控制。

电动机使用日本伺服会社制作的 DME34B37G61A。一般地,对相扑机器人脚旋转,使用 DME34B37G18A,将传动比变大,传动比使用 61:1 减速

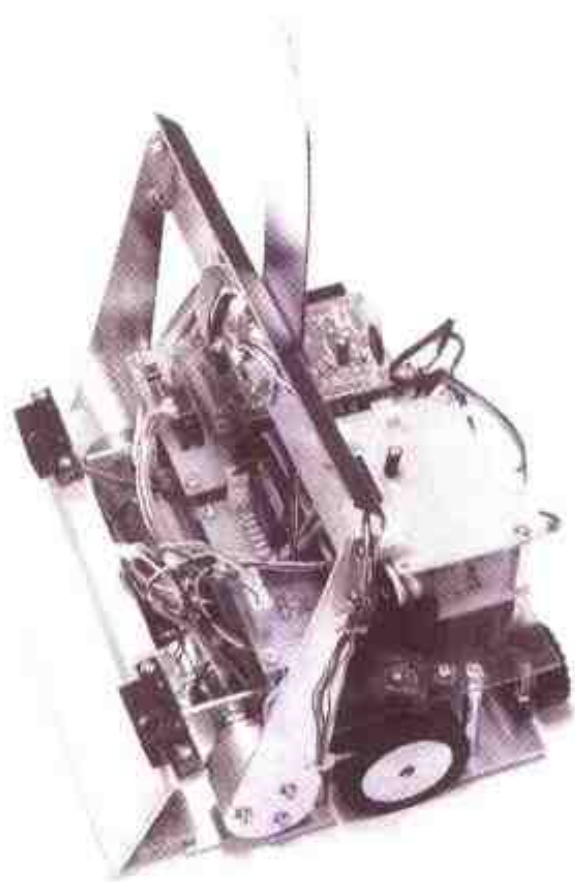


• 照片 8.21 臂的结构

装置。使用臂时输出轴仅能旋转 120° 左右。作为臂的速度尽管是这种传动比也是足够的。这个电动机额定转数为 62.2r/min , 为了转动 120° 需要 0.35s , 是没有问题的速度。

电位器使用 $5\text{k}\Omega$ 左右可变电阻, 其轴径使用和电动机输出轴相同 ($\Phi 4\text{mm}$) 即可。

电动机驱动电路使用如图 8.9 的电路。机器人相扑手臂的安装如照片 8.22 所示。



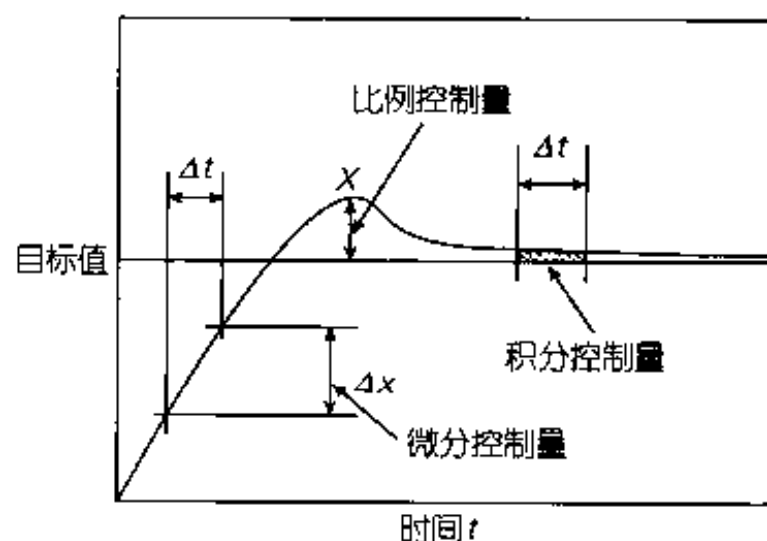
• 照片 8.22 安装臂的位置

8.4.3 臂的实际控制

这里使用 PID 控制, 试进行臂的控制。PID 控制使用在各个领域, 它是个比较简单的方法。PID 控制也是古典的控制方法, 然而现在仍然使用在

很多场合。

这里有关 PID 控制简单地进行叙述。



• 图 8.16 PID 控制的图形

控制就是对某目标位置为基准调节实际位置的过程。机器人臂的情况是以臂的位置为目标, 对应于这个位置迅速、准确地调整, 这就是它的目的。使机器人臂既摇晃, 又不偏离位置进行控制。为此, 改变电动机的正转、反转或转速等。这时, 为了求出在电动机中流过多大的电流最好的作法是, 通过计算求控制量。将其用 P(比例控制)、I(积分控制)和 D(微分控制)的考虑方法进行控制。

1. 比例控制(P)

由目标值给出对偏离比例的控制量。

控制量为 $P = k_p * p$

$p = (\text{现在的位置} - \text{目标值})$

现在的位置由电位器读取电压值。在与目标值的差变大时, 控制量即这里就是电流, 使流过电动机的电流变大。换句话说, 就是使 PWM 的脉

宽变大,使其流过大电流。

这里 k_p 为比例系数,让机器人臂运动,反复按照本能,习惯试行,屡次错误而逐渐得到适应。对到达标的位置需要花费时间时,将这个值变大即可。

2. 积分控制(I)

例如,电动机打算抬起臂,而由于臂的重量,往往在稍微下降一点儿的位置才完全稳定。这里如何处理呢?将现在位置进行几次积分即可。所谓积分控制就是根据积分获得的值决定的控制。偏离积分值 $i = (\text{由过去偏离的某时间的积分值})$

控制值 $I = k_i * i$

k_i 为比例系数

通过追加它,可以使目标值偏离值变小。

3. 微分控制(D)

微分控制是对决定机器人臂精确位置的重要控制。机器人臂摇晃时,加入微分控制时恰好稳定。

对于比例控制,例如从上向下变化臂,超过目标值时,将电动机反向旋转。这时另一支臂也超过目标值,于是迅速反转电动机。反复操作它时臂

摇晃。为了抑制臂摇晃进行微分控制。对这个微分控制量,是从上往下变化臂时的变化量,即控制使用微分值。从上往下快速变化时,稍微保持控制量进行控制。

$d = (\text{过去的位置} - \text{现在的位置})$

只要设控制量 $D = k_d * d$ 即可,

k_d 为比例系数。

将上述 3 个控制合并起来阐述时,控制量变为:

$$C = P + I + D = k_p * p + k_i * i + k_d * d$$

这就是 PID 控制。后面只要将其编程序即可。将这个处理,使用中斷试进行程序设计。

程序

驱动电动机插件板连接在微型计算机端口 PA、PB 上。电位器连接在 AD 转换器 AN0 上。

臂控制通过一定时间(每 5ms)中斷进行。对主程序只要仅仅指示臂的位置即可。通过步长开关值变更臂的位置。

臂控制在上述 PID 控制中使用 PD 控制。

chapter

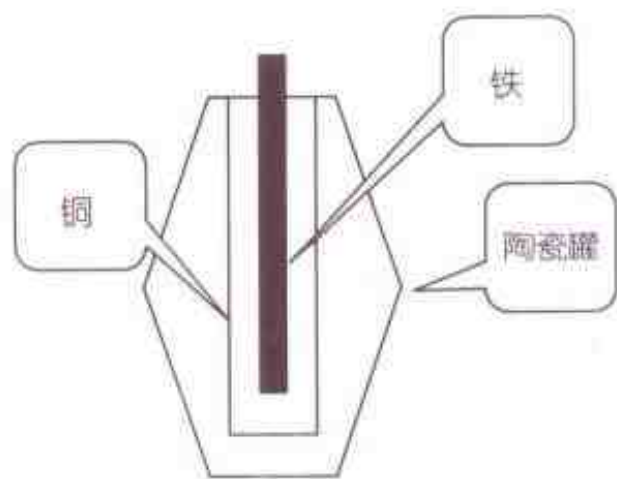
9

机器人的能量

一般机器人竞赛，使用蓄电池作为机器人的能源。充分了解蓄电池的特性之后再制作机器人，可以制作速度更快、力量更强的机器人。

9.1 蓄电池

在 2000 年前,蓄电池好像使用在铝铁磁合金上镀金,在陶瓷罐中放入铁和铜的电极开发电池,在电解液中使用 winvinegar 时,据说会产生 0.4V 左右的电压(参见图 9.1 所示)。



• 图 9.1 初期的蓄电池

以后,蓄电池继续进化,现在笔记本电脑、手提电话、电动汽车等所有领域已经使用起新型蓄电池。特别是蓄电池的大容量化和 IC 模块小电力化的进步,手提电话和可听频率关系的发展,蓄电池变得经久耐用。

9.1.1 碱电池

碱电池(参见照片 9.1 所示)是日本微型计算机圣杯机器人竞赛中指定的蓄电池。但如果没有充分地了解它的特性就用于机器人制作,会由于 CPU 的清 0 或电压的变化,使机器人车的速度发生变化,而造成机器人不

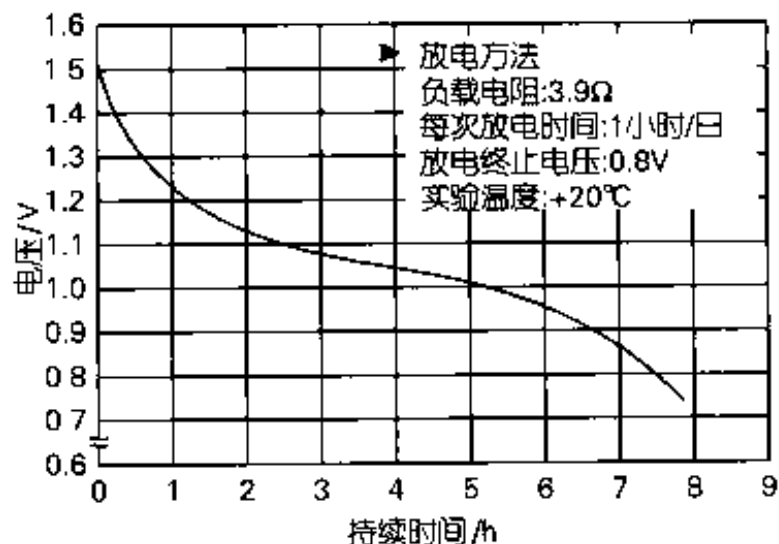
能熟练地动作。



• 照片 9.1 碱电池

碱电池相对镍-镉电池来说,电流不大爱流动,所以必需下功夫选择电动机。

锰电池或碱电池放电电压随着电池的消耗成 S 字型曲线下降(参见图 9.2 所示)。最初,电压发生急剧下降。流过大电流时下降非常显著,在微型计算机圣杯机器人等中,由于开始行走速度和最后行走速度是变化的,因此必需进行速度控制或电压控制。



• 图 9.2 放电特性

9.1.2 镍-镉电池

镍-镉蓄电池(参见照片 9.2 所示)在机器人竞赛中是理所当然使用最多的电池。因为它能够流过大电流,并且可以充电,另外放电特性也好。对必需的机器人电源来说,相扑机器人等具有良好的特性。另一方面,由于它是利用化学反应的蓄电池,所以它产生存储电能的效应。另外,进行快速充电时,老化也快。



• 照片 9.2 镍-镉电池

在正极为镍氧化物、负极为镉化合物,电解液使用氢氧化钾水溶液。镍-镉电池的工作原理,通过下面的反应式记述它的化学反应。

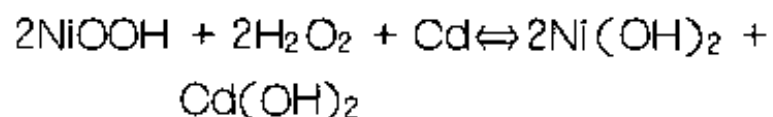
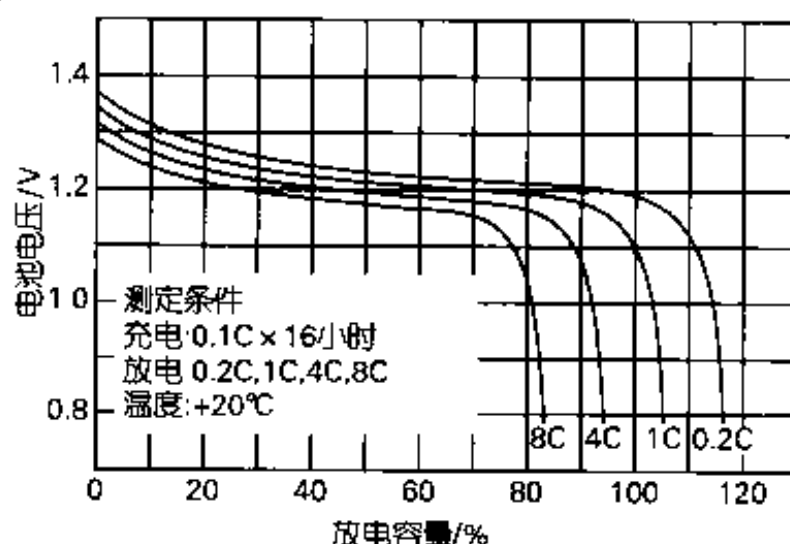


图 9.3 为镍-镉电池的放电特性,而流过大电流时,电压在相当容量部分稳定,这是它的特征。

使用蓄电池盒等情况,由于流过大电流,接触电阻变大,电压变低成为微型计算机清 0 的原因,所以必需注意。使用模型店经营的蓄电池盒(照

片 9.3 所示)即可。另外,充电器有各种各样,所以希望同时购回。但是,尺寸和重量必须能装载在机器人上。



• 图 9.3 镍-镉电池的放电特性



• 照片 9.3 无线遥控用蓄电池

9.1.3 镍-氢电池

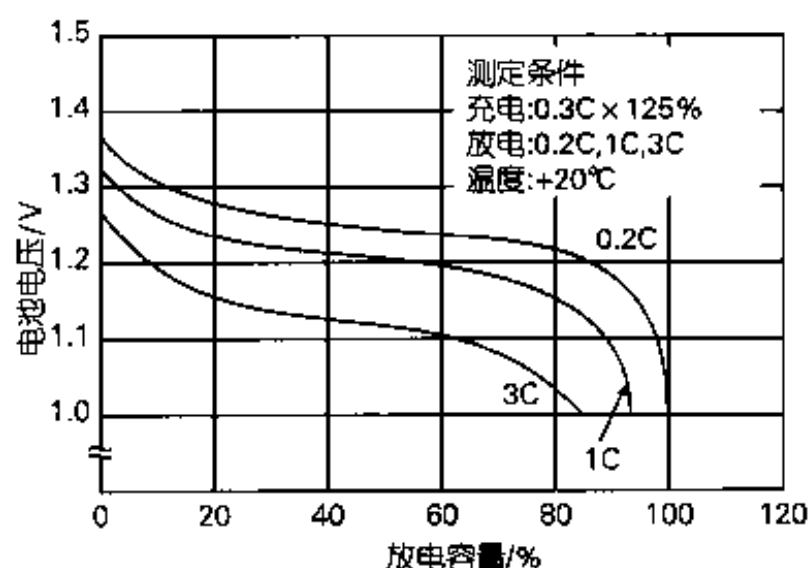
镍-氢电池(照片 9.4 所示)正极为镍氧化物,负极为氢吸留合金,电解液使用氢氧化钾。与同样尺寸的镍-镉电

池相比,镍-氢电池是2倍以上高能密度的蓄电池。



• 照片 9.4 镍-氢电池

其放电特性在图 9.4 中表示。最近机器人领域也已经使用起这种电池。

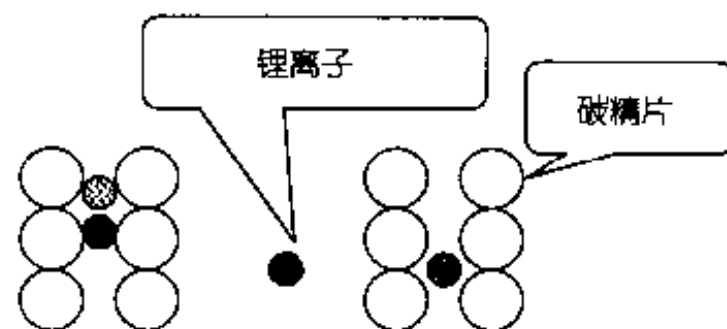


• 图 9.4 镍-氢电池的放电特性

9.1.4 锂离子二次电池

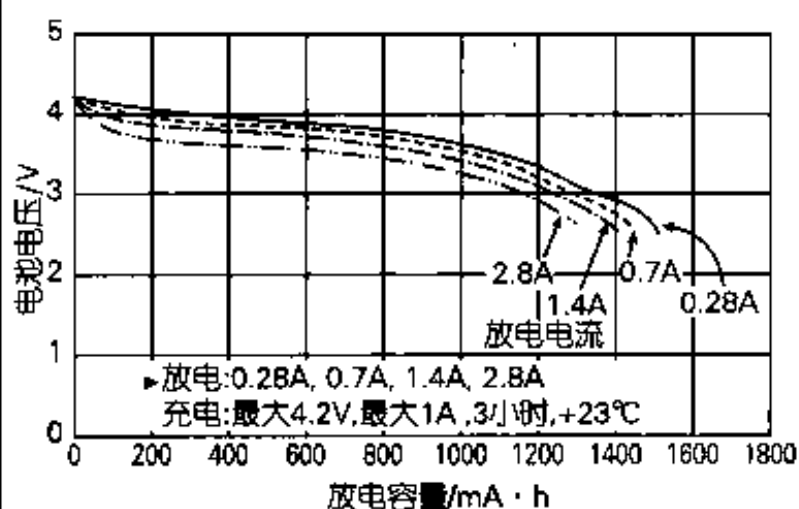
锂离子二次电池(锂离子蓄电池)是在英国发明的,而发明人是日本人,这种电池未使用化学反应,下面简单地介绍一下工作原理。如图 9.5 通过移动锂离子正、负电极,进行充、放电。锂产生 $-3V$ 电压,钴产生 $+1.2V$ 电

压,因此,锂离子二次电池最大产生 $4.2V$ 电压。



• 图 9.5 锂离子蓄电池的原理

图 9.6 是锂离子蓄电池的放电特性,由图可知,它具有初期电压变化很小,随着放电容量增加电压慢慢地降低的特性。



• 图 9.6 锂离子蓄电池的放电特性

1) 锂离子二次电池的特征

①充、放电效率高,接近 100%,其它电池由于化学反应发热,所以效率低。

②没有存储效应,存储效应通过化学反应由不同的金属产生,所以必需对其充电。由于锂离子二次电池不通过化学反应,所以不产生存储效应。

②在原理上它反复使用的次数无

限大,对应几千,几万次的充、放电。

④能量大。

充电时要注意的事项。施加电压进行快速充电时,锂离子全部集中在碳精片方面,不可以向上述那样地充电。因此,充电必需慢慢地进行,需要 2 小

时左右。电极使用稀有的陶瓷技术,在这个素烧瓷的尺寸上确定充电时间。

在初期时快,以后进行沉长地充电,慢慢地充电效率高。还有,在购买锂离子二次电池的同时,也购买专用充电器即可。

9.2 充、放电器

镍镉蓄电池是利用化学反应的电池,所以产生存储效应。曾经手提电话也使用过镍镉电池,另外,手提个人电脑或录像机也是一样。反复充、放电时,在放电没有充分放完的状态,再进行充电时,即使这种状态也不能再放电,剩下的所谓记忆蓄电池,会很快完全放电。这曾经是蓄电池的寿命变短的原因之一。最近更换了锂离子蓄电池,所以手提电话的蓄电池寿命充分延长,另外能量密度也高,充一次电保持的时间也变长。

照片 9.5 为模型店出售的镍镉蓄电池充电器,它附有放电器的功能。一旦蓄电池使用完了,用放电器放好电,由存储效应产生的问题就没有了。



• 照片 9.5 放电器附带充电器

用放电器进行长时间放电,平均每一节电池的电压到 1.0V 左右进行完全放电,这样存储效应没有了,返回原来的状态。

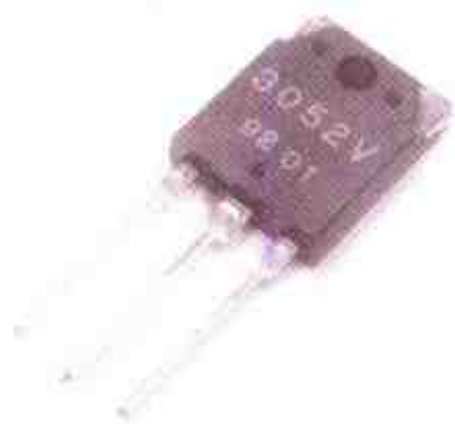
9.3 CPU 用的 5V 电源电路

微型计算机或传感器等一般使用 5V 电压,所以必需准备 5V 电源。将 4 只镍镉蓄电池串联连接时大体为 5V 电压,可以驱动微型计算机。另外,有三

个引脚的调节器 IC 模块,电压从 7.2~14.4V,可以方便地产生 5V 电源。

为了从驱动电动机蓄电池中获得 5V 电源供给 CPU 等使用,采用了 3 个

引脚的调节器。对这个调节器使用 3052V, 其外形在照片 9.6 中表示。

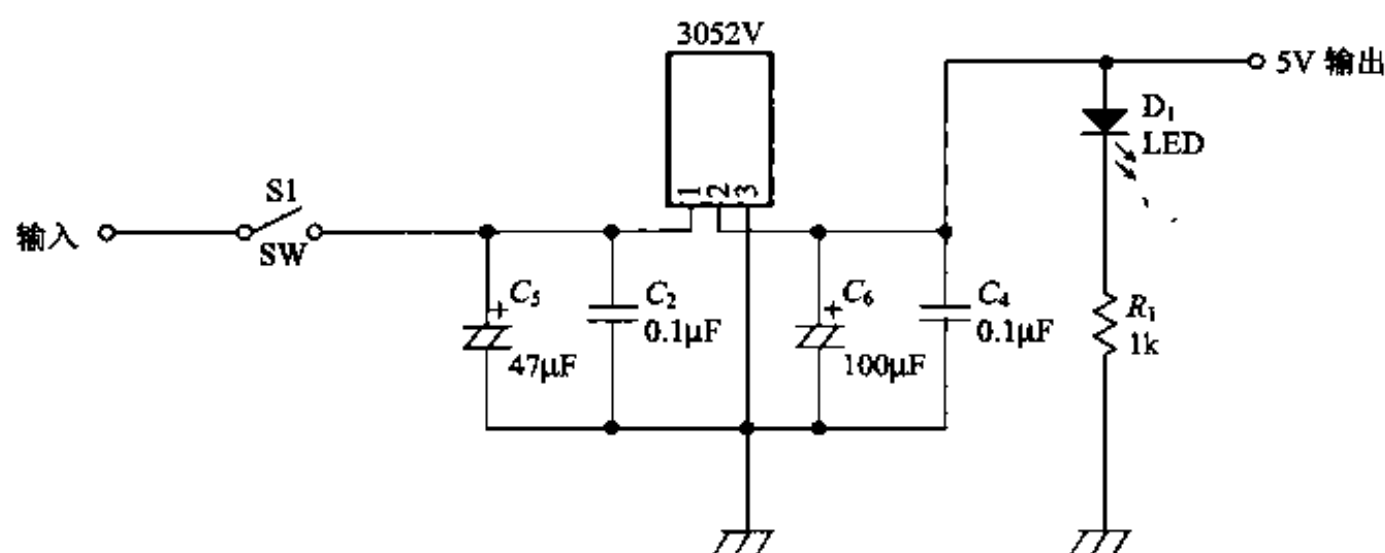


• 照片 9.6 3052V 调节器

这个 IC 模块工作时输入/输出电

压差为 1V 以下, 很难受输入电压降低的影响。例如, 由于驱动电动机产生蓄电池电压降低。在电动机中流过大电流, 由于内部电阻的作用, 蓄电池电压即使降低, 也不能降到 6V 以下, 可以维持 5V 输出电压。

7805 三个引脚调节器等工作时, 输入/输出电压差和其相比大, 是可以安心地使用的 IC 模块。另外, 由于输出电流也可以使用到 2A, 所以非常方便。对在机器人中使用不要散热片, 其电路图在图 9.7 中表示。



• 图 9.7 5V 电源电路

• 使用 DC/DC 转换器

DC/DC 转换器通过开关产生高电压, 使其整流就可以获得需要的电压, 这个电压可以上升, 也可以下降, 如照片 9.7 封装成一体, 市场有出售, 所以利用它即可。包括各种各样电压量程, 照片 9.7 是 Bellnix 社的产品, 信息可以从以下网址获得。

<http://www.bellnix.co.jp/>



• 照片 9.7 市场出售的 DC/DC 转换器

chapter 10

相扑机器人的 制作方法

相扑机器人对形状和结构而言自由度多，参加大会竞赛的机器人具有各种各样的结构。为了在大会上继续获得胜利，要求有适应于各种机器人的更先进的技术。本章将叙述有关相扑机器人的控制及各种机器人的结构。并列举出基本型相扑机器人制作实例。

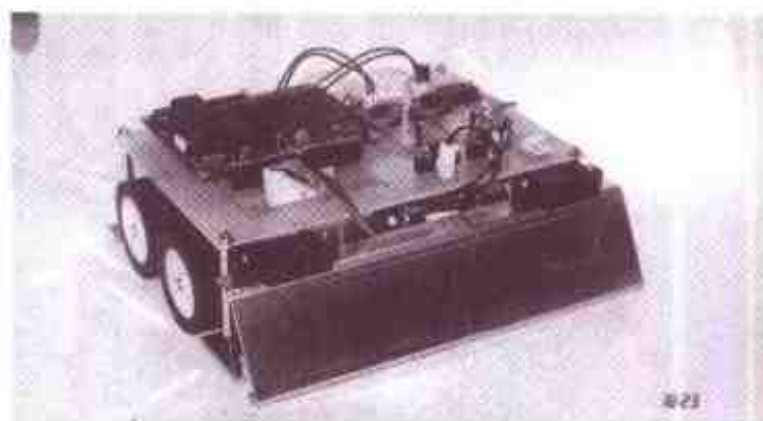
10.1 机器人的构成

目前出场机器人大会的自律型相扑机器人力士,装有蓄电池,用电动机和轮胎运动。为了使机器人不从摔跤场上跌落下来,安装有检测摔跤场的传感器和检测对手的传感器。而且,由于是自律型机器人,所以装载微型计算机,由传感器的输入控制电动机。



• 照片 10.1 2 轮驱动型机器人

车身一般如照片 10.1、照片 10.2 所示,有 2 轮驱动型和 4 轮驱动型的两种类型。最近又出现了 6 轮驱动的机器人。



• 照片 10.2 4 轮驱动型机器人

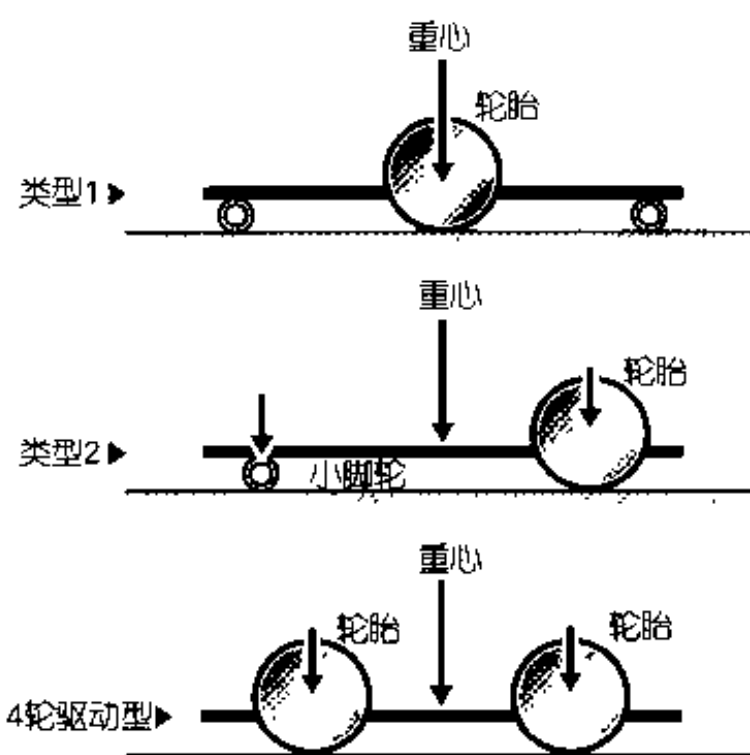
10.2 相扑机器人的车身

目前出场的相扑机器人如图 10.1 所示,由图可见,有 3 种类型的机器人。类型 1 机器人的重量全部作用于轮胎上。类型 2 重量的一半作用于轮胎上,剩下的一半作用于小脚轮上。因而,对于类型 2 的车身,在轮胎与摔跤场的摩擦阻力减少一半的同时,小脚轮的阻力增加了,不能充分发挥电动机的转矩。在装载 CPU 插件板和蓄电池的情况下,在不得不使用类型 2 时,要进行重量的调整,使重心尽量靠近车轮附近。另外,也能经常看见 4

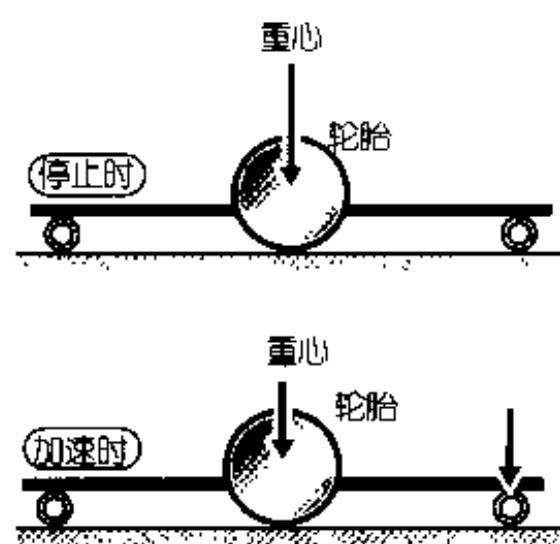
轮驱动型,然而结构复杂,同时受重量和空间的限制。还有在 4 轮驱动中,也有使用齿轮、链条结构,用一个电动机驱动 2 个轮胎的类型。

为了提高加速度性能,有提高电动机的转矩、轻量化,提高轮胎的夹钳(grip)和降低旋转阻力等方法。特别是对小脚轮,尽可能使其不受负载作用,这是理想状态。在加速时,如图 10.2 所示,小脚轮受负载作用,旋转阻力增加,同时轮胎的分担负载减少,从轮胎到路面之间传送转矩就不能很好

地完成。另外,在旋转过程中,由此会引起旋转角稳定性的恶化。



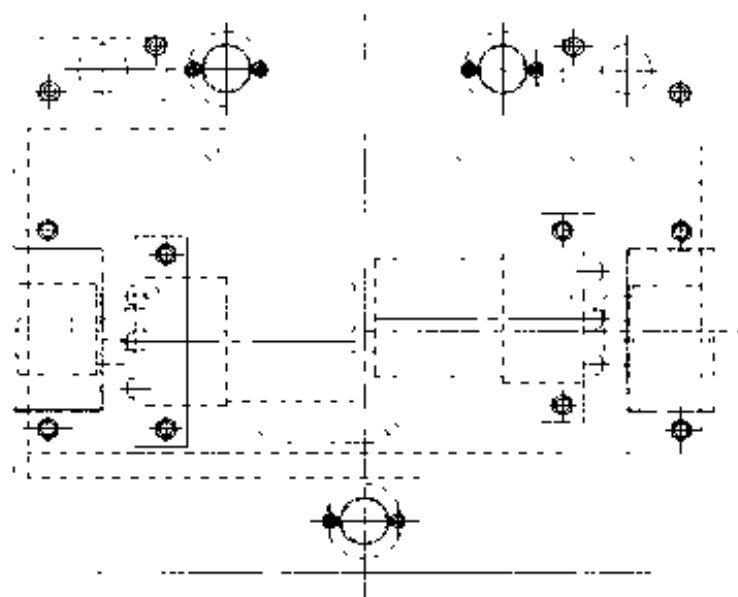
• 图 10.1 相扑机器人车身



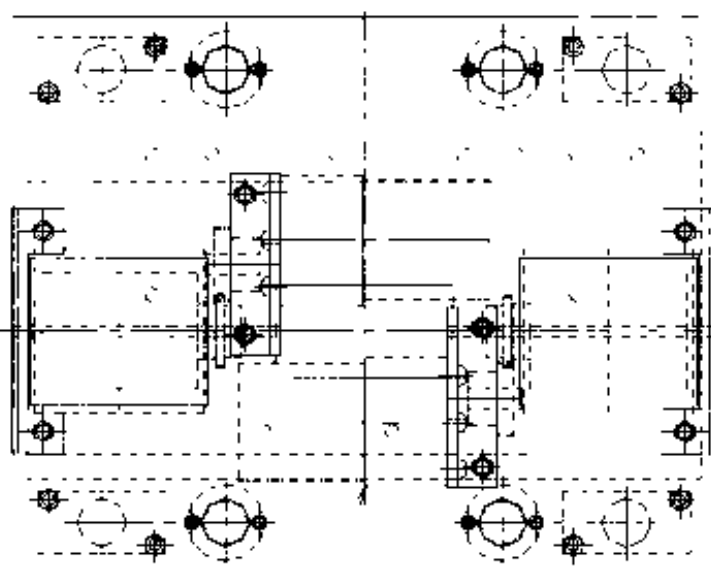
• 图 10.2 加速时负载平衡

实际车身的设计实例

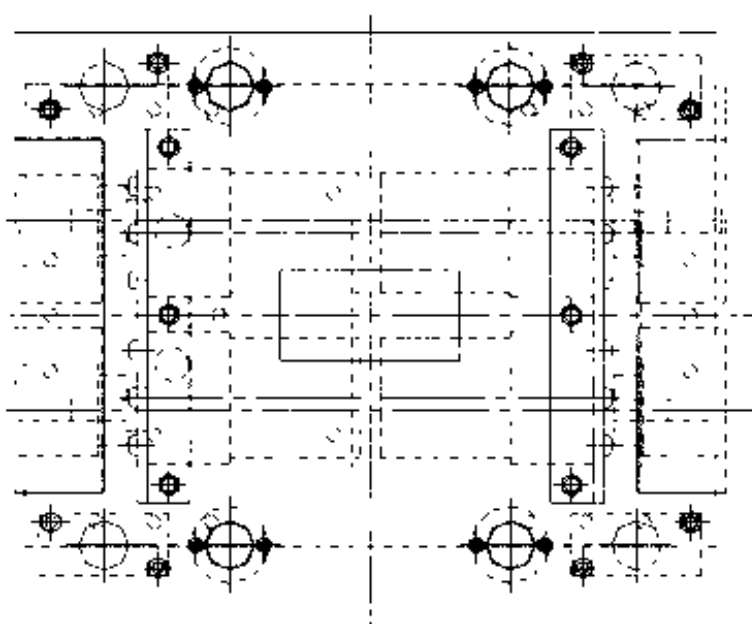
图 10.3、图 10.4、图 10.5 列举了具有代表性的车身设计实例。可参考这些实例设计自律型的相扑机器人。



• 图 10.3 相扑机器人的车身(2轮型)



• 图 10.4 相扑机器人的车身(齿轮型)



• 图 10.5 相扑机器人的车身(4轮型)

10.3 车板(电话卡)的安装方法

对相扑机器人,如何钻到对方的下面是非常重要的,在相扑机器人领域,将车板这样的部件称为裙板(skirt)。使用这个裙板钻到对手的下面。这个裙板的制作水平左右胜负。

在最近相扑机器人大会上,有使用磨出锋利刃的金属板和薄不锈钢片的部件,然而这样相当危险。本来制作的是欣赏性机器人,结果变成了浑身是伤的机器人。

这里叙述有关正确裙板的制作方法。这个裙板的制作尽管在机器人的制作过程中花费很大的功夫也不算浪费。这仅仅就是意味着有钻到对手下面的可能。

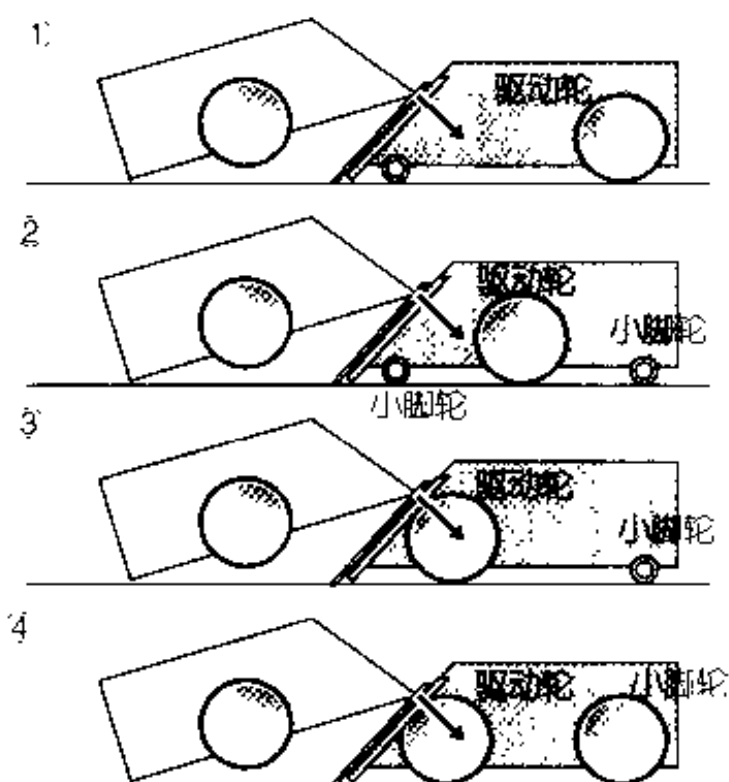
在相扑机器人领域,一段时间内称为“微米战争”时期。这种战争就是如何使裙板变薄,钻到对手的下面的战争。这场战争的结束是因为出现了具有臂结构和吸引型等新的技术的机器人。尽管如此,不容质疑的是,钻入对手下面还是重要的。

1. 为什么钻入对手下面就有力量?

为什么钻入对手下面就有力量呢?下面简单说明它的力学原理。

如图 10.6 的①所示,被对手抬起的机器人轮胎完全悬起,不能把动力传给摔跤场。因而,应该能够轻易地把它推出场,但是,最近吸引型机器人变得多起来,所以就不能那么轻易地

将其推出场。必须把从对手(机器人)受到的力高效地传到驱动轮上。



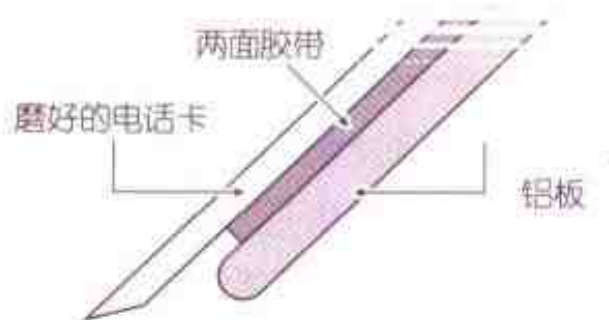
• 图 10.6 钻入对手下面车身的 4 种情况

图 10.6①中对手的力传到小脚轮上,几乎没有传到驱动轮上。图 10.6 的②和③把驱动轮设置在前端时,对手的力就加到驱动轮上了。这样,就可充分利用对手的力推进比赛的进展。这确实是相扑机器人的妙趣。可以说,从这一点看,图 10.6④中的 4 轮驱动型机器人起到了这种效果。

以上,讨论了钻到对手下面的同时,把对手的力高效地传到驱动轮的结构,运用这一结构即使对具有很强吸引力的机器人也可以充分对抗。

2. 裙板的构造

在充分理解以上力学原理的基础上,可以进行裙板及机器人的制作,在图 10.7 中表示裙板的结构,相关问题将在以下的 content 中讲解。



• 图 10.7 裙板的结构

3. 电话卡的磨制方法

准备两张电话卡,试对撞一下看,由于电话卡厚度相同,两者撞在一起不动。

将一方磨削加工看看。用砂纸从宽的一端顺序加工下去,精加工 2000~3000 遍即可。

将两张电话卡再次对撞一下看,磨过的和没磨的电话卡的差别就明显出来了,这就是微米战。如何使顶端变薄,并且使之能随摔跤场凹凸不平做弹性起伏,这就是顶端部分的制作要点。

可是,磨过量时,电话卡的强度就下降,顶端易坏,所以要注意。磨削几张叠加在一起试验是非常重要的。

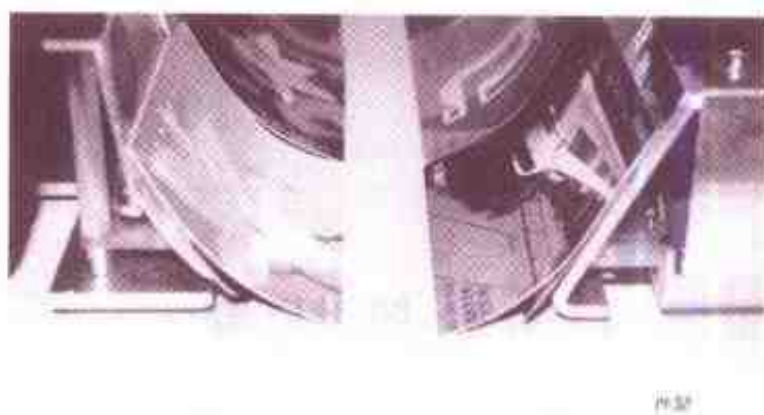
4. 在机器人上安装时的注意事项

电话卡和铝板之间的安装用双面胶带,这是为了能把损坏的电话卡迅速替换下来。

安装电话卡时,关键的问题是铝板对于摔跤场的倾斜角度和接触的位

置,即用多大的角度使电话卡接触地面。

参考照片 10.3,试研究有关安装角度和电话卡接触地面的问题。



• 照片 10.3 裙板的安装(两台机器人)

而且,最重要之点是电话卡的切剪处理。在相扑机器人的比赛中,机器人几乎不相互从正面碰撞,从斜面碰撞时会如何呢? 对这种情况的研究也是非常重要的。

将电话卡的一个面倾斜,试试对战的结果。由于在电话卡上带有圆弧半径,用圆弧部分对战很脆弱。相反,制作出能瞄准这里进攻的模式也是重要的事情。

将如照片 10.4 所示电话卡棱部锐利地剪下即可。



• 照片 10.4 电话卡的剪切

5. 电话卡的加固

最近也有把不锈钢等材料用于机器人裙板的,所以比赛时有将电话卡完全切下来的时候。为此,必需将电

话卡用铝板巧妙地加固好,将铝板最大限度地伸向摔跤场,为了不损坏摔跤场希望进行好半圆面处理(参见图 10.7)。

10.4 吸引的方法

10.4.1 吸引风扇

曾经在相扑机器人的顶部安装飞机用的螺旋桨,旋转螺旋桨将机器人压紧在摔跤场上,在上升到天花板上这种遥控方向机器人出现的时候,让大家非常吃惊。他们认认真真的(遵守规则)开始比赛,开场的时候使风扇旋转起来。但是,还没等风扇发挥作用就被对手推出场外。后来把风扇装在机器人的内部,在开场前就让风扇旋转,但产生了强烈的风扇噪声和振动,也明显地暴露出机器人在动作。富士软件 ABC 的优点就是按照标准,不严格限制机器人的制作,所以标准变了,即标准灵活了。为了有效利用新技术机器人,将作出认可改变标准的姿态。

可是,担心认可这种吸引型,会对相扑机器人的发展起消极的影响。在机器人领域内,仍然希望应该由软件控制自律性或动作或速度等,这是考虑的原因。

最近,出现了一种新的机器人,它可以一边轻快地吸引,一边快速地运动,发现了对手的弱点突然袭击,这正

好解除了我们的担心。今后参赛的相扑机器人,期望朝此方向发展。

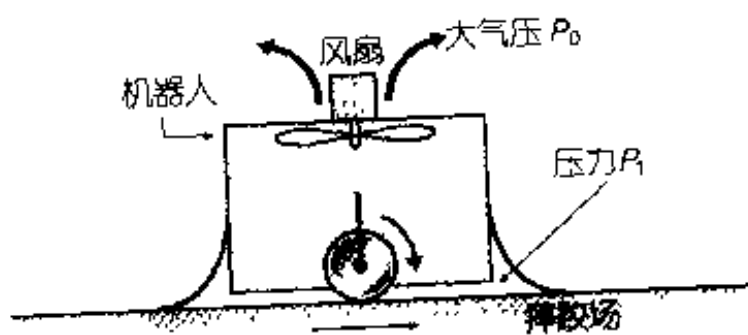
在最近的大多数相扑机器人中,装载了吸引装置,在表 10.1 中,列出了在最近的机器人大会上吸引装置的有无情况,作为最近的倾向,可以看出空吸泵型机器人的增加,另外,大多数获奖者几乎都装载了几个吸引装置。

表 10.1 在最近机器人大会上吸引装置的有无情况

大会名	型式	吸引		没有吸引
		风扇型	泵型	
1998 年全日本高中生大会	无线方向遥控型	10	13	9
	自律型	7	16	9
1998 年全日本大会	无线方向遥控型	19	24	21
	自律型	17	30	17

1. 吸引为什么就有力量

在图 10.8 表示吸引的原理。由于风扇旋转吸出空气,机器人内的压力(P_1)降低。其与大气压 P_0 的压差($P_0 - P_1$),使机器人内的面积 S 的积形成力作用于轮胎上。由此,黏着力提高,并且和电动机产生的力组合在一起,压力就增大。



• 图 10.8 吸引的原理

但是，与此同时旋转阻力增加。另外，密封部分的负压也变为阻力，必需提高电动机的转矩。因而，使用相同的电动机时，必要选取大减速比的皮带轮。为此，一般吸引型的机器人速度慢。为了快速，就必需使用大输出功率的电动机，电动机相应就大而重，于是就超重了。

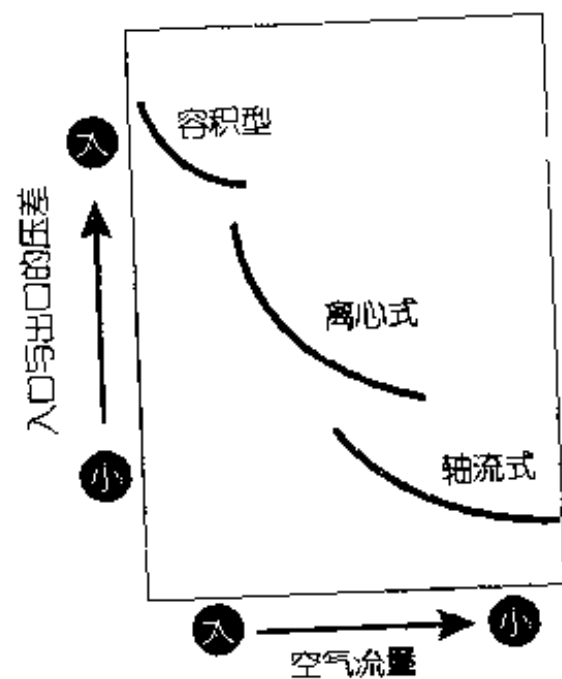
如何选择这些平衡，成为制作吸引型机器人的决定因素。

2. 吸引风扇的结构

图 10.9 示于吸引机的特性。容积型能够取得大的压力差，然而流量小。另一方面，轴流式流量能取得大值，而压差没能起作用。离心式处于两者之间的位置。

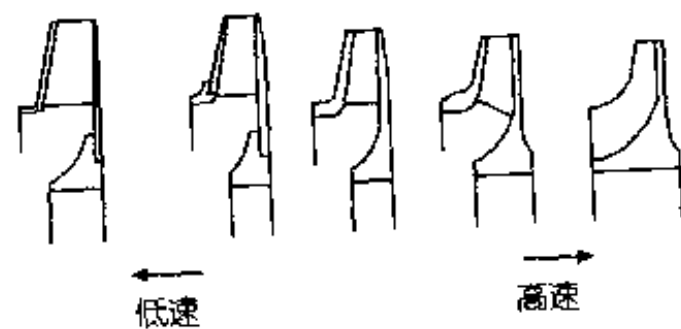
轴流式风扇气流是轴向的，应用于飞机的螺旋桨及美发用的风筒等各种领域。离心式风扇气流是相对于旋转轴在离心力的方向流动，所以吸尘器的风扇一般使用这种方式。

对相扑机器人，某种程度上需要降低压力，所以多使用离心式。大部分机器人使用从吸尘器上卸下用过的气扇，然而好像也出现了一部分自作风扇的机器人。



• 图 10.9 吸引机的特性

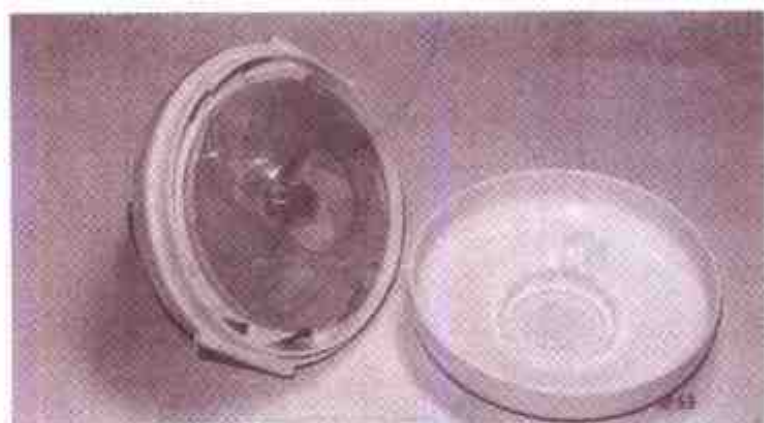
即使是离心式风扇，也要有与风扇旋转适应的风扇形状，如图 10.10 所示。对低速型和高速型，形状各不相同。低速型使用于吸尘器；高速型在汽车的涡轮增压器等中使用，用几十万转的高速旋转。因为电动机的转速是几千转，所以可以说是低速型。



• 图 10.10 离心式叶片的形状

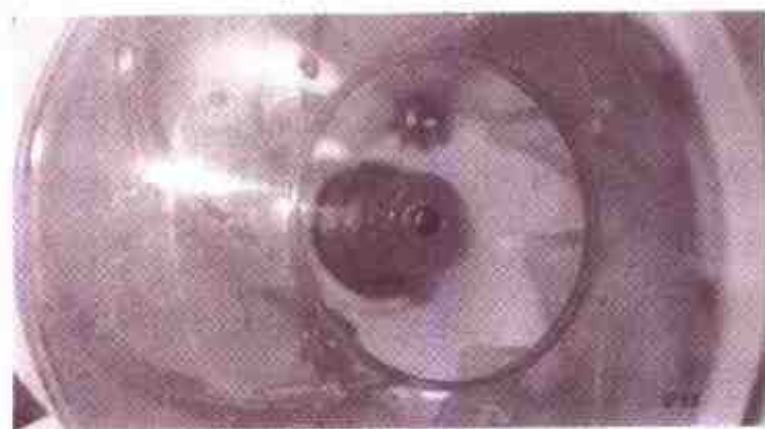
拆开便携式吸尘器看时，可见到如照片 10.5 所示的风扇。有使用充电式的和家庭用 100V（日本民用电为 100V）电源的。这里，使用带有用蓄电池充电型吸尘器的风扇。带有防护罩的风扇只要单独安装在机器人上即可，然而，防护罩和吸尘器主体成为一

体,卸下风扇,防护罩就没有了,所以购买时要注意。



• 照片 10.5 风 扇

通过提高风扇的转速可使负压变大,从而可以增加大气压差。由此,可以提高吸引力。在无线遥控商店有各种各样高速旋转的电动机出售。一般无线方向遥控用电动机和同形状的电动机也使用在吸尘器的风扇上,所以,这种电动机也有使用的可能。但是,如照片 10.6 所示,风扇被压入电动机中,卸下它非常不容易。要有损坏最初带有的电动机的心理准备,抓住电动机轴,拔出风扇即可。照片 10.7 为安装着无线方向遥控的电动机。转速超速时,由于离心力的作用,往往损坏风扇,所以必需十分注意。在风扇的旋转方向上,希望一定要安装防护罩。



• 照片 10.6 压入风扇



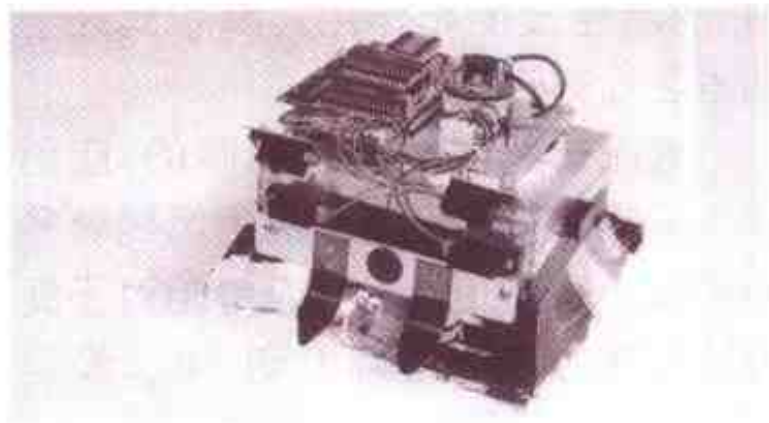
• 照片 10.7 更换电动机

3. 密封方法

为了提高风扇的吸引力,不仅提高转速,而且密封性的提高也很重要。对空气的密封有以下方法:

• 外周密封方式

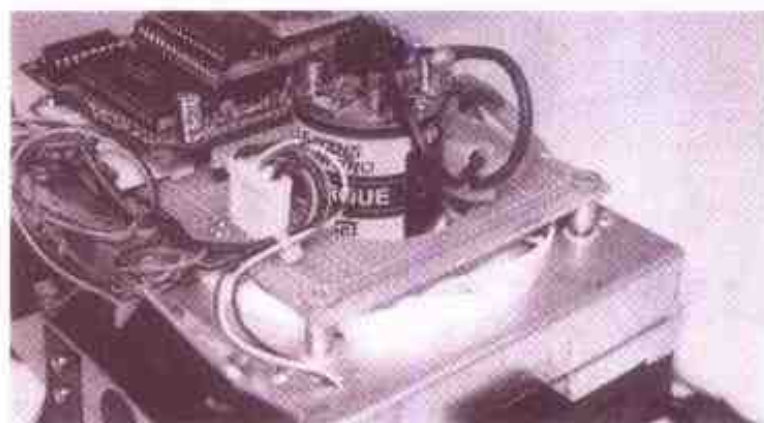
第一种方法是在最外周的密封,这种方式一般用于带有吸引风扇的相扑机器人(如照片 10.8、照片 10.9 所示)。这种密封部分也起着裙板的作用,所以具有吸附力以及能够不被对手攻入的用途。但是,与对手交战时,密封部分容易受伤,所以必需频繁地更换。



• 照片 10.8 周围密封方式

全周密封使用电话卡等,由于既有柔软性,摩擦系数又小,即使是吸引也能平滑地运动。另外,这种裙板的柔软性变为挡板,即使机器人主体被

激烈地挤压的时候,吸引密封也能很好地保持,不会发生气体泄漏。



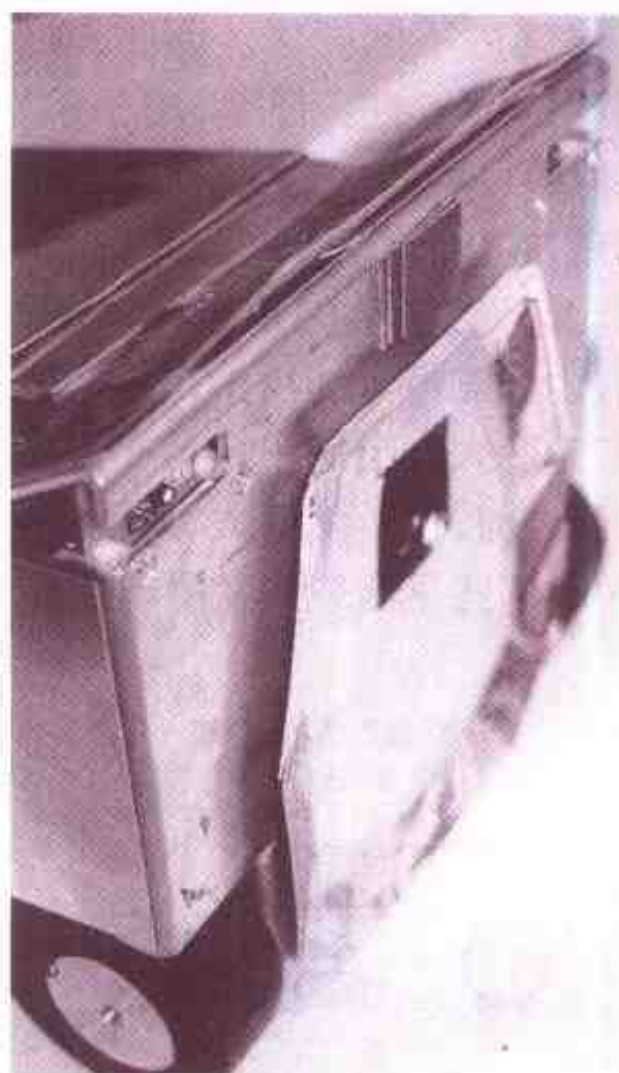
• 照片 10.9 风扇的安装位置

• 内部密封方式

对外周密封方式,被对手攻入下部时,气体会发生泄漏,吸引力完全下降。为此,将密封部分稍微保持在内部进行,即便被稍微攻入,也没有破坏到密封,仍然具有密封的结构(如照片 10.10 所示)。刚好就如鲶虎鱼的吸盘一样,这些部位也用电话卡等制做密封部分。

但是,这种情况会由于吸引面积减少而使吸引力下降。因而,为了得到和外周密封相同的吸引力,就必需增加吸引风扇的负压。

除此之外,还有各种各样精心设计的机器人。例如,进行压力计测控制的机器人。如果压力下降,也许就从摔跤场上脱开,被举了起来,这时要有为了能从对手那里逃脱那样的控制功能。



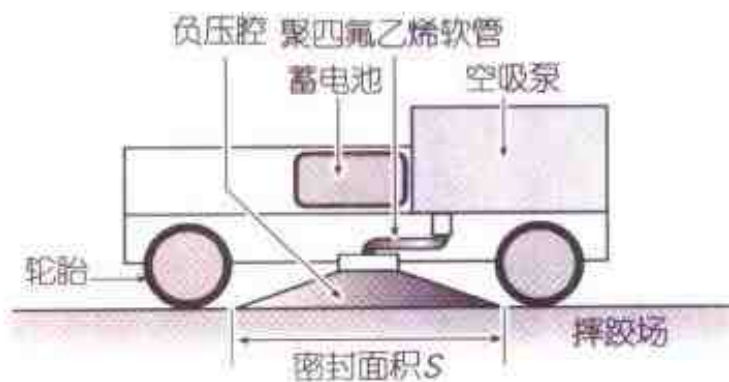
• 照片 10.10 内侧密封(机器人的里侧)

10.4.2 容积泵

说起泵,有观赏鱼用的气泵和抽水用的水泵,以及空气压缩机等等。这些都是使泵的出口压力提高。相反,入口处的压力下降,如果使用这种方法就可做出真空泵。

图 10.11,表示使用空吸泵的机器人原理。一般情况,泵型与风扇型相比,空气的流量小,而由于可得到负压(参照图 10.9 所示吸引机的特性),想得到同样的吸引力时,减少负压腔的密封面积就可解决(负压腔的密封面积 \times 负压=吸引力)。另一方面,因为空气的流量小,所以密封性不好时,压

力下降就达不到要求。因此,必须确保足够的密封性。但是,由于机器人紧贴摔跤场地地面,想要提高密封性,摩擦力就增加。相反,机器人的运动就变坏。因而,泵型即使和同等的扇型相比,也必需将电动机的转矩(驱动轮胎的力)变大。电动机容量相同,将传动比变大时,转矩就变大,然而速度变慢,所以这些平衡关系变得非常重要。



• 图 10.11 装载空吸泵机器人的原理

1. 各种各样的吸引泵

对扇型泵,利用容积变化进行吸引的泵称为容积泵。作为汽车的增压装置被利用的有罗茨型、里校鲁姆型等,然而考虑到对结构复杂的相扑机器人不相称,所以这里省略说明。

这里介绍可以装载在相扑机器人上,并且在市面上出售的隔膜式和活塞式泵。

• 隔膜式泵

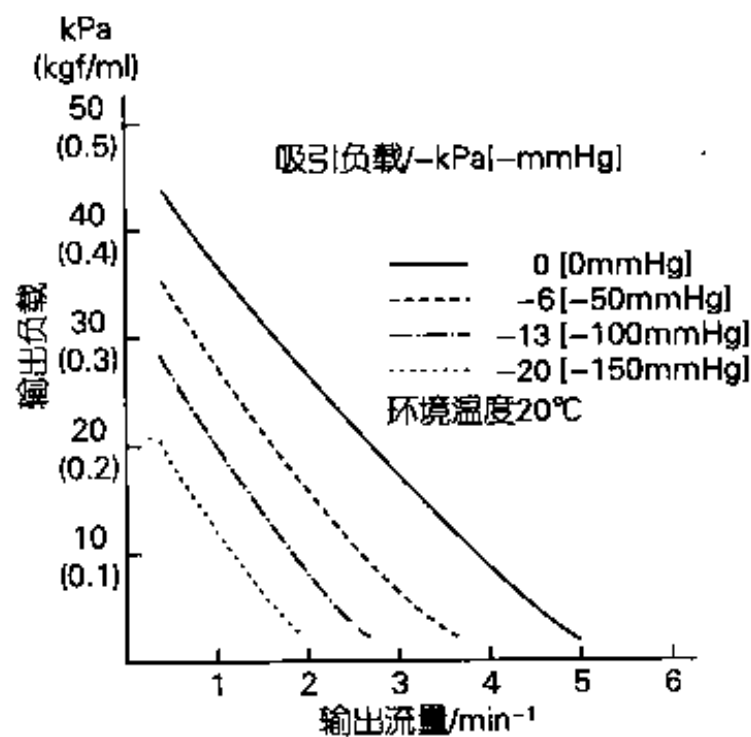
可用于相扑机器人那种尺寸的隔膜式泵在日本国内正在广泛出售。由于隔膜是用橡胶制作的,所以没有足够的刚性,不能够承受太大的负压(如照片 10.11 所示)。



• 照片 10.11 隔膜式泵

这种泵的重量为 340g,真空度最大可达 250mmHg 以上。当确保 100mm²的密封面积时,就可以计算得到 20kg 的吸引力,然而,实际上由于密封性等的原因,不能得到 100%的吸引力,但从效果看还似乎不错。

图 10.12 示出隔膜式泵的特性。输出负载是大气压,可考虑为零,所以吸引负载由输出流量决定。即提高密封性,减少输出流量就能够增大负压。

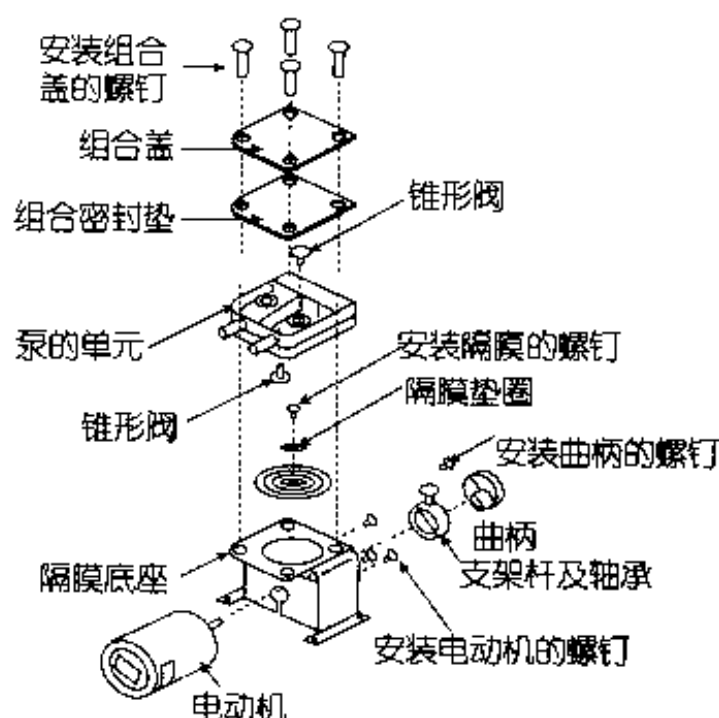


• 图 10.12 隔膜泵的特性

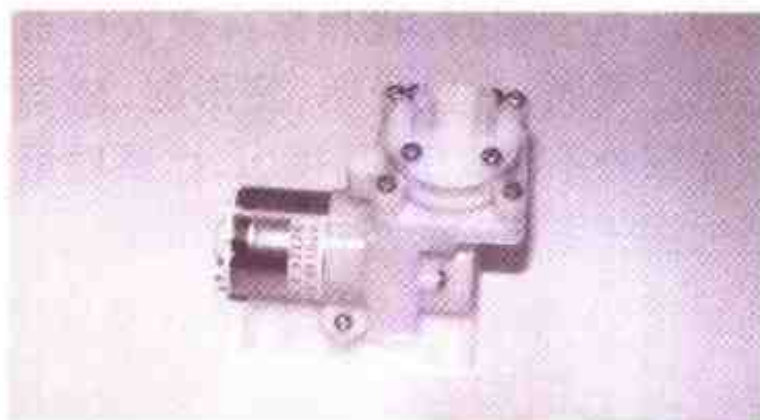
这种泵的结构如图 10.13 所示,通过上下移动隔膜产生负压。

照片 10.12 所示隔膜式泵 (DP0140) 由 Medo 产业株式会社出售。小型的负压可达到 400mmHg, 最

适合于相扑机器人。用 12V 直流电压工作, 重量 190g, 是轻型泵。



• 图 10.13 隔膜式泵的结构



• 照片 10.12 隔膜式泵(DP0140)

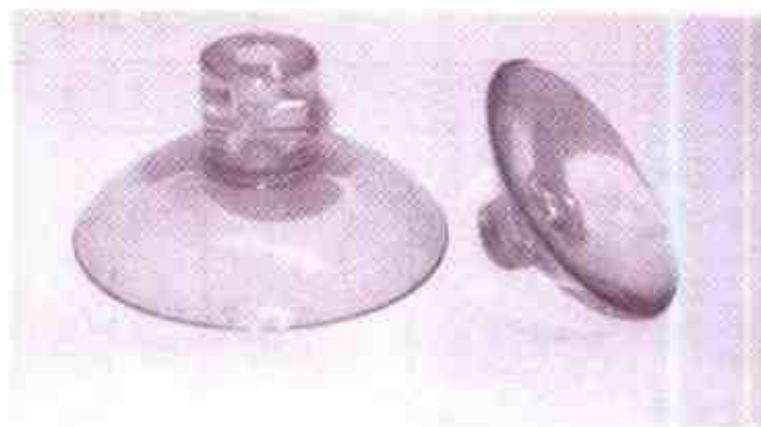
Medo 产业株式会社网址如下:
<http://www.medo.co.jp/>

2. 吸引密封

对负压部分使用如照片 10.13 所示吸盘即可。吸盘密封性好, 被举起时密封面积就增加, 吸引力增大。但是, 必需根据泵的能力和电动机的转

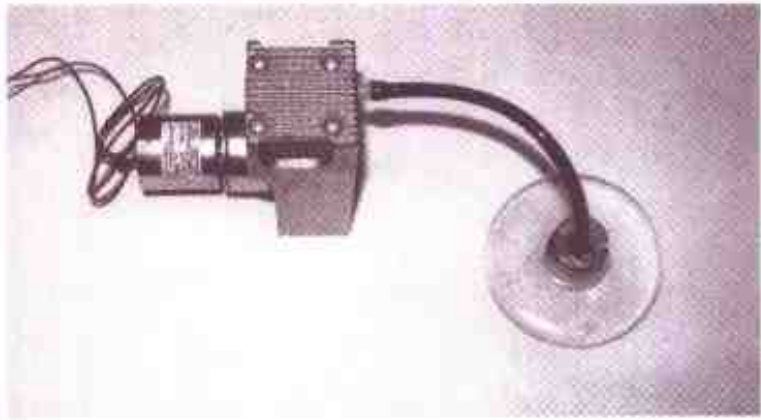
矩选定其尺寸。

在机器人大会上, 也发现了内部不用空吸泵, 只用吸盘的吸引机器人。然而, 吸引力随时间降低, 所以不大可能向前运动。



• 照片 10.13 各种吸盘

如照片 10.14 所示, 把软管黏接在吸盘上用泵吸引。吸盘必需牢牢地固定在机器人上。



• 照片 10.14 与泵的连接

3. 吸盘的滑动

吸引泵型机器人吸附在摔跤场的部分使用大型的橡胶制作的吸盘,把机器人放在摔跤场上,驱动泵若提高真空度,机器人当然就贴在摔跤场上。但是,根据“开始”的指令,按起动按钮,过了 5 秒钟,机器人竟还纹丝不动。

若考虑一下,觉得也是很自然的事情,橡胶做的摔跤场和吸盘间产生的摩擦力相当大,即使吸引型机器人电动机的功率适当,能够克服这个摩擦也很不容易了。即使运动了,电动

机大部分的能量被用于强行拖曳吸盘,也就拿不出推出竞争对手所需的能量。

不是一般性质的吸盘,是用聚四氟乙烯加工成的蝶形物吸附在摔跤场上。然而,蒸镀聚四氟乙烯不是外行所能做的技术。只要滑动就吸住,具有这个功能的部分不管什么材质都行。即可以理解为即使是普通的吸盘也无所谓。可将滑动的材质依次收集起来,评论比较手感、柔软性、耐久性、加工容易性和摔跤场相容性等(参见表 10.2)。

从成本上看上述几点,我想可以使用购物袋(尼龙制作)及家具滑板这种水平就可以了。购物袋如果破了,可以很随意地更换,对于练习比赛很方便。对于实战,我想家具滑板是最好的。然而,一旦受伤、变形时,就有可能保持不住真空度,所以,比赛后最好检验一下它的必然状态。

表 10.2 吸盘的比较

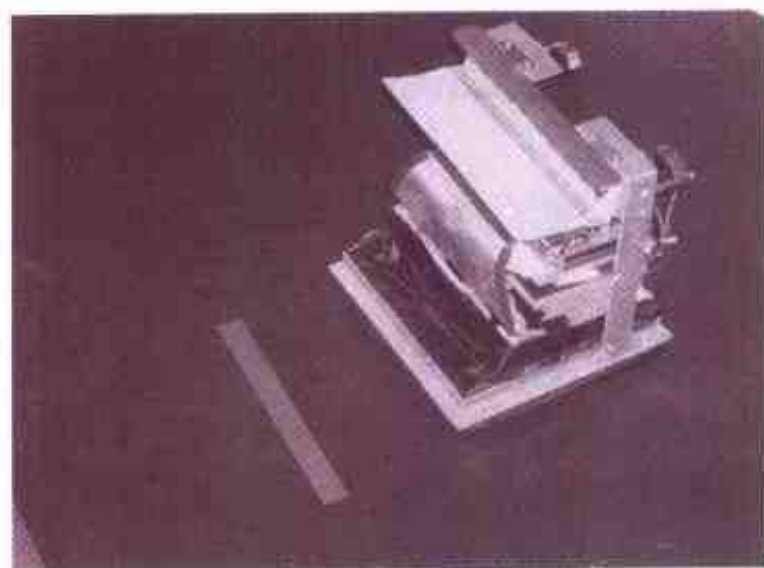
	手感	柔软性	耐久性	加工容易性	与摔跤场的相容性
聚四氟乙烯板	○	○	△	○	○
家具滑板	○	△	○	○	○
购物袋	○	○	△	○	○
纸	△	○	×	○	△
蜡	△	△	×	×	×
塑料板	○	×	○	△	○
氟喷射	○	○	△	○	○

10.5 相扑机器人臂的结构

与以前相比,参加相扑机器人大会的机器人多数具有臂结构。但是,不那么容易使用那个臂结构的功能。最近一段时期,正发展使臂快速起落,巧妙钻入对手下面等功效。

为了有效地运用臂的功效,必需整理好它的各种特征。尤其是如何快速起落臂成为要点。另外,要知道由于安装臂而产生的优点和缺点。不仅仅是安装上臂,而且也要考虑如何使用臂作为进攻的手段。

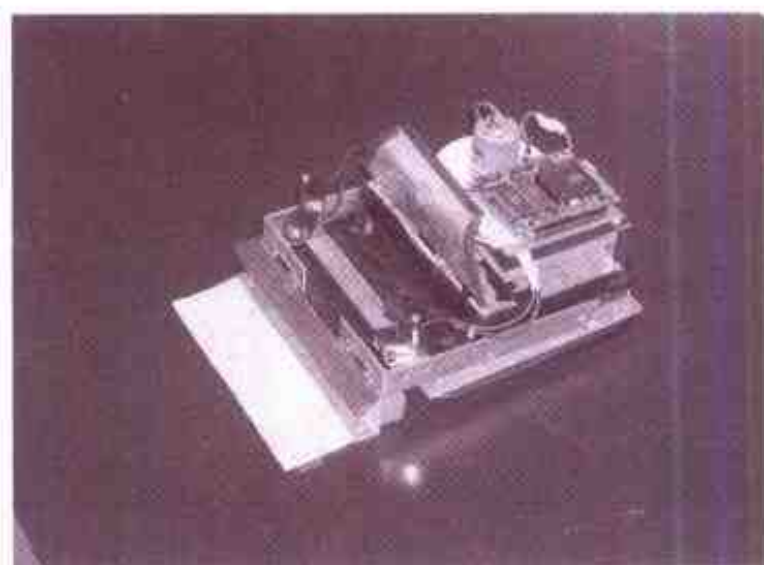
照片 10.15 为安装臂的机器人在开场前的状态,把臂举在上面,大小必须在 $20\text{cm} \times 20\text{cm}$ 的范围之内。照片 10.16 是开场时臂刚落下的状态。这里,如何快速落下这个臂,而形成攻击点是最重要的。



• 照片 10.15 具有臂结构的机器人(臂落下前)

下面,就有关特征性臂的结构和

它的起落方法进行叙述。



• 照片 10.16 具有臂结构的机器人(臂落下后)

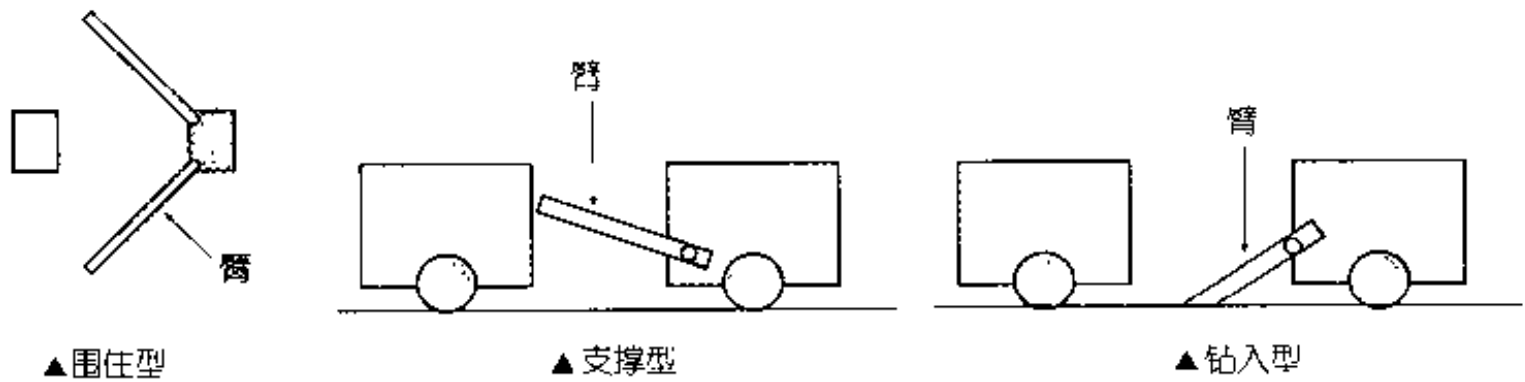
1. 安装臂的目的

臂有各种结构,其目的也各不相同。臂的结构大体上如图 10.14 所示。下面叙述有关这些臂的用途、安装技巧及进攻方法。

安装臂的目的大致为以下几种:

- ①用臂围住对手推出:围住型;
- ②用臂支撑对手,高效推出:支撑型;
- ③将臂钻入对手下面,推出:钻入型。

表 10.3 中表示了第 10 届全日本相扑机器人大会上臂的安装比率。大体上 $40\% \sim 50\%$ 的机器人装载上述任意一种臂。也有很多将这些臂组合装载的机器人。下面讲解有关这些结构的功能。



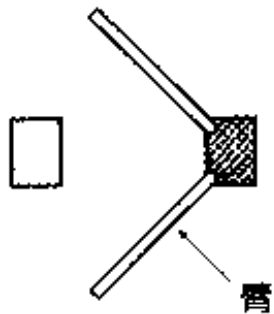
• 图 10.14 臂的各种结构

表 10.3 第 10 届大会臂型机器人的比率

臂的类型	无线遥控型	自律型
钻入型	33%	16%
支撑型	7%	13%
围住型	0%	19%

2. 围住型

如图 10.15 所示,它是将臂大展开、把对手围住。这种类型多为自律型机器人。因为没有必要安装检测对方的传感器。以前,也只有直线型机器人装有该装置。这里抓住对手并将它推出场地,所以围住型机器人的硬件和软件都很简单。对无线遥控型,人们边看对手边操作,所以这种类型的臂没有太大效果,好像目前也不使用了。



• 图 10.15 臂结构的要点

未装 CPU 的自律型机器人曾经胜过装有 CPU 的机器人。开场的同时打开两臂,沿着摔跤场滴溜滴溜地旋转,它是直线追踪机器人。侧面的臂一抓住对手,通过臂的作用自动地面向对手。之后,仅用锋利的裙板将对手推出场地。像这种围住型臂可以准确地抓住对手。如果机械部分没有问题,捕捉确实不靠电气部分控制。

结构如下。要点是臂的长度、落下的角度及落下的方法。这边臂还没落下时,对手已跑到它臂的外侧,所以完全处于无防备状态。

因而,这种机器人的进攻能力主要体现在臂伸展的程度和抵御下面进攻的能力。

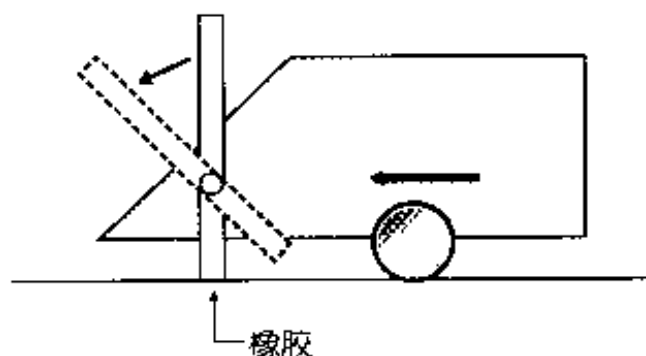
对于臂的落下方法有各种各样的观点。

最简单的方法就是仅仅把臂立着,让机器人后退一步的方法。之后,靠重力落下。另外,也有让机器人旋转的方法,这是使用离心力的方法。任何一种方法都是由于一会儿后退、一会儿前进,时间使用过多,给了对手逃跑的时间。因而,必需在开场同时就落下臂上下功夫。将臂先用弹簧拉

着,开场前将它锁定好,开场的同时解除牵引。这个方法也有各种各样的形式。

(1) 利用机器人向前运动力的方法

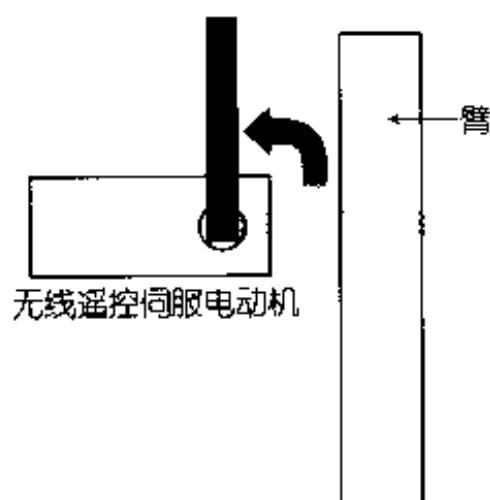
利用这个原理,落下锁定装置。在开场前固定臂,这时臂的下部接触摔跤场。开场的同时机器人向前运动。利用这个力,落下臂。由于这个动作合并机器人的运动中,所以可以使臂快速落下(如图 10.16 所示)。



• 图 10.16 利用机器人运动的力

(2) 利用无线方向遥控伺服电动机

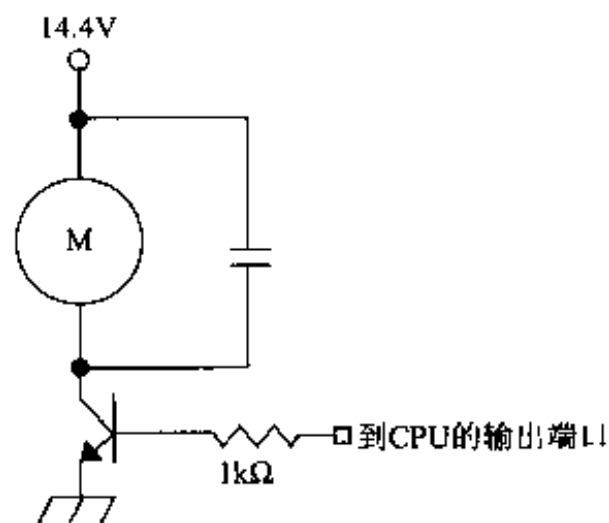
如图 10.17 所示,让无线方向遥控伺服电动机旋转,解除锁定。对无线遥控型如果通道数允许,是比较简单的。然而,对自律型就稍微麻烦,有关这种方法在后面叙述。



• 图 10.17 由伺服电动机解除锁定装置

(3) 用电动机解除锁定

让电动机旋转代替图 10.17 中的部分,解除锁定。同汽车的门锁定是同样原理。开场时同时让电动机旋转,解除锁定。也根据电动机的种类,尽管自律型也可用图 10.18 所示简单的电路使电动机旋转,但使用带有减速齿轮机构即可。



• 图 10.18 电动机控制

(4) 电磁阀

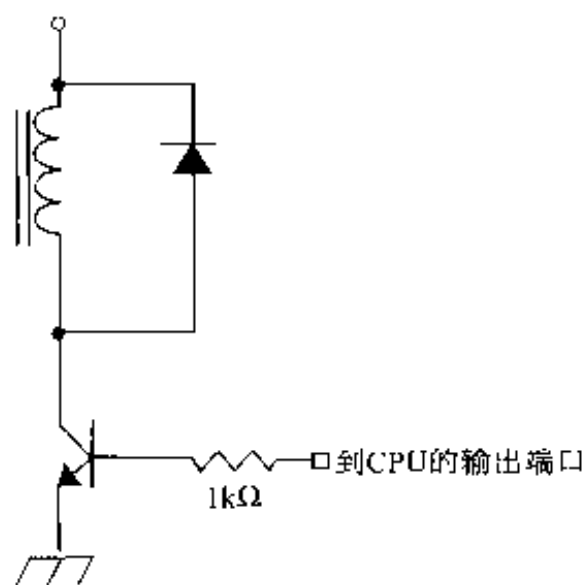
开场前先用电磁阀吸住臂,起动的同时解除锁定。这需把臂平衡制作好,尽量降低电磁阀必需的能量。电路如图 10.19 所示。用自律型进行竞赛时,开场前将电磁阀接通(ON),开场的同时断开(OFF)电磁阀。之后,靠弹簧力和重力落下臂。

对于围住型臂,也能使其具有臂之外的功能。

(1) 安装接触传感器

由于安装了传感器,可以朝向对手的方向。由此,可以一直推出对手,高效地传送力。通过对手触及臂,输入开关信号进行判断。关于接触传感

器,在“机器人控制”杂志 No. 4(欧姆社)中有详细的讲解,希望参照阅读。



• 图 10.19 电磁阀控制

(2) 安装欺人眼目的装置

在自律型相扑机器人中,在腕部空白处安装能够检测对手的装置时,对手会将它当作敌人判断,面向那个方向,以对手正在认真地控制这是前提。臂伸向侧面,所以对手决定面向侧面。从而,如果推它,就能将对手从侧面推出。

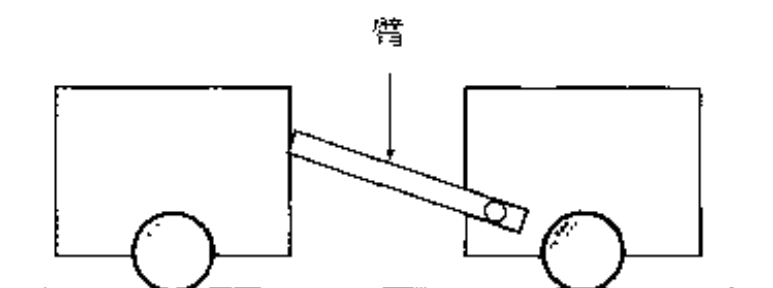
作为这种进攻方法,制作好不用检测信号的模式,也可以和对手从正面作战。

(3) 把裙板安装在臂上

由于在臂上安装裙板,对于对手,是斜向钻入裙板。一般对于斜面,裙板防御不充分的时候多。因而,钻入对手下面的机率较高。

3. 支撑型臂

图 10.20 示出了安装支撑杆时的力学效果情况。利用对手的推力,抬起对手的向时,提高了自身轮胎的黏着力。



• 图 10.20 安装支撑杆时的效果

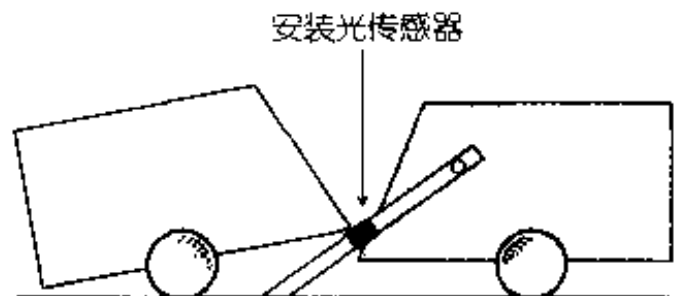
这种情况重要的是支撑杆尽量瞄准对手的上部。通过腕部的角度,可以把对手的力加到自身的轮胎上。

但是,如果对于避开了这个支撑杆,就什么效果也没有了。作为预防的办法,在杆的顶端贴上橡胶,提高黏着力。必需在挂住对手上下功夫。另外,要与对手的结构相符,调整臂的高度等等的办法也很重要。

4. 钻入型臂

钻入型臂是把臂钻入对手薄弱的环节,一口气推出场外。不能很好地钻入对手的下面时,也有多次重复钻入对手的自律型机器人。这种类型有很多把臂的顶端磨尖。必需注意不能损坏摔跤场。

另外,在无线遥控型中可以看见这样的一种方法,一声不响地等着对手开到臂上,对手一旦开到臂上,一口气把它推出去。为了用自律型完成这种动作,如图 10.21 所示;要安装判断对手开上来了的光传感器。



• 图 10.21 钻入型臂

10.6 机器人的制作实例

10.6.1 2 轮型机器人的制作

那么,是否能制作出任何事情都会做的机器人呢?这是个问题。

制作像本田那样的用二脚步行机器人或索尼那样的宠物机器人是不可能的,尽管我们希望看到能使其作出相似的动作,但是,这只是制作者的心情罢了。然而,对于步行机器人,制作步行机构也是很困难的,并且其设计程序更加困难。

NIGHT RIDER 的网址

夜间骑士乘坐的机器人车称为 KITT(Knight Industry Two Thousands)。这个机器人车具有自动驾驶、自动追踪及回避冲撞功能,也可以说话。按照夜间骑士的指示,作各种各样的事情。在紧急时通过 KITT 的判断,使驱动器完全逃脱危险。

“夜间骑士”是美国 1982 年开始放映的最受欢迎的电视节目,在日本 1987 年左右才开始放映这个电视节目。对此较感兴趣的读者,请参照以下网址读取。

<http://www.geocities.com/Hollywood/Academy/3367/>

10.6.2 机器人的概念

对机器人首先确定让它做什么事情?

对相扑机器人,战胜对手是最重要的。为了战胜对手,力量要强,钻入到对手的下面、发现对手等等,需要各种各样的条件,而这里首先把基本的条件作为满足机器人的要求。

1. 基本功能

- (1) 不能从摔跤场跌落;
- (2) 能够发现对手;
- (3) 追赶对手。

再加上各位制作者,要在如何制作出强有力的机器人上下功夫。

2. 附加功能

为了使相扑机器人竞赛有欣赏性,让机器人事先产生一些特色。

- (1) 对机器人,通过传感器给与指示,使其能行走到分界线;
- (2) 一旦战胜对手,能举起胜者的名字,绕摔跤场一周。

将如上所述各种程序考虑好,以后编进去。

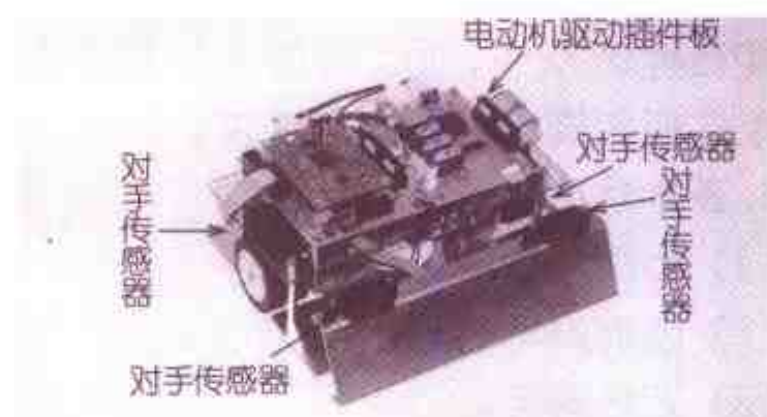
3. 适用动作

- (1) 摇头表示不愿意;
- (2) 点头表示同意;
- (3) 迅速靠近;
- (4) 发现对手的弱点;
- (5) 如果推没成功,就逃跑。

还有其它各种动作,而在使用基本功能的基础上用简单的软件补加适用的动作。

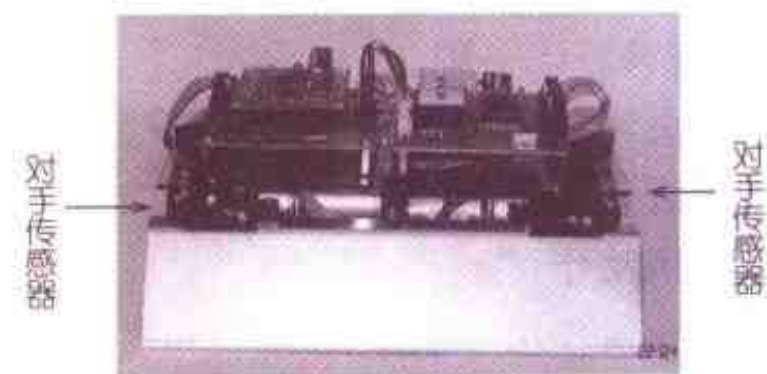
10.6.3 机器人概述

照片 10.17 是机器人的总体图。旋转脚是 2 轮驱动。将 2 台电动机左、右排列安好,改变它的转速使机器人动作。这就是全部的机器人竞赛的基本条件。



• 照片 10.17 机器人的总体图

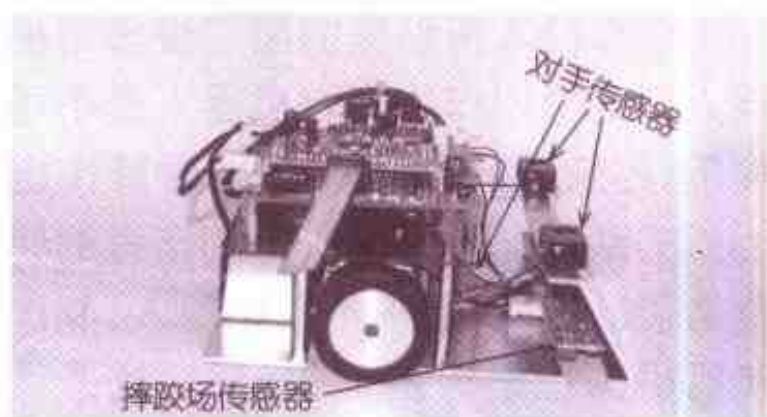
照片 10.18~10.22 是从各个方面看到的制作完成机器人的照片。它使用 4 个对手传感器、4 个摔跤场传感器。



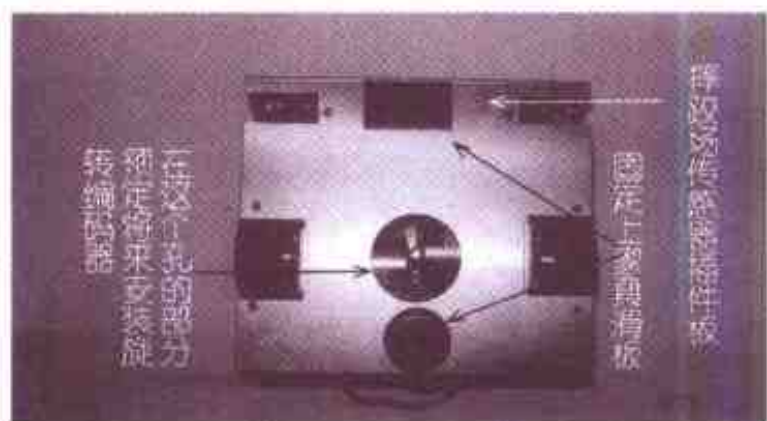
• 照片 10.18 机器人正面



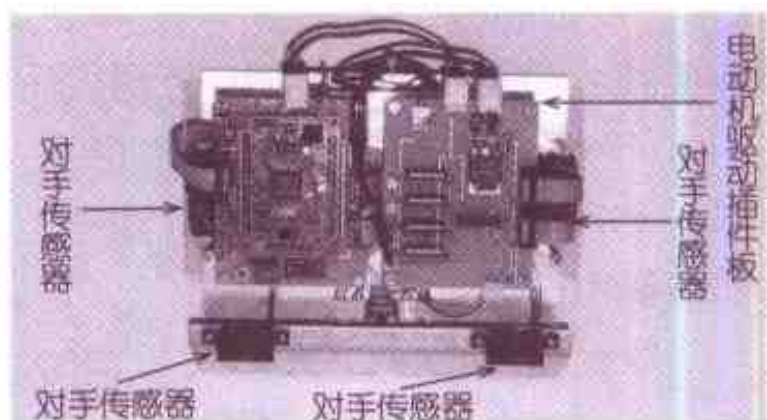
• 照片 10.19 机器人后面



• 照片 10.20 机器人侧面



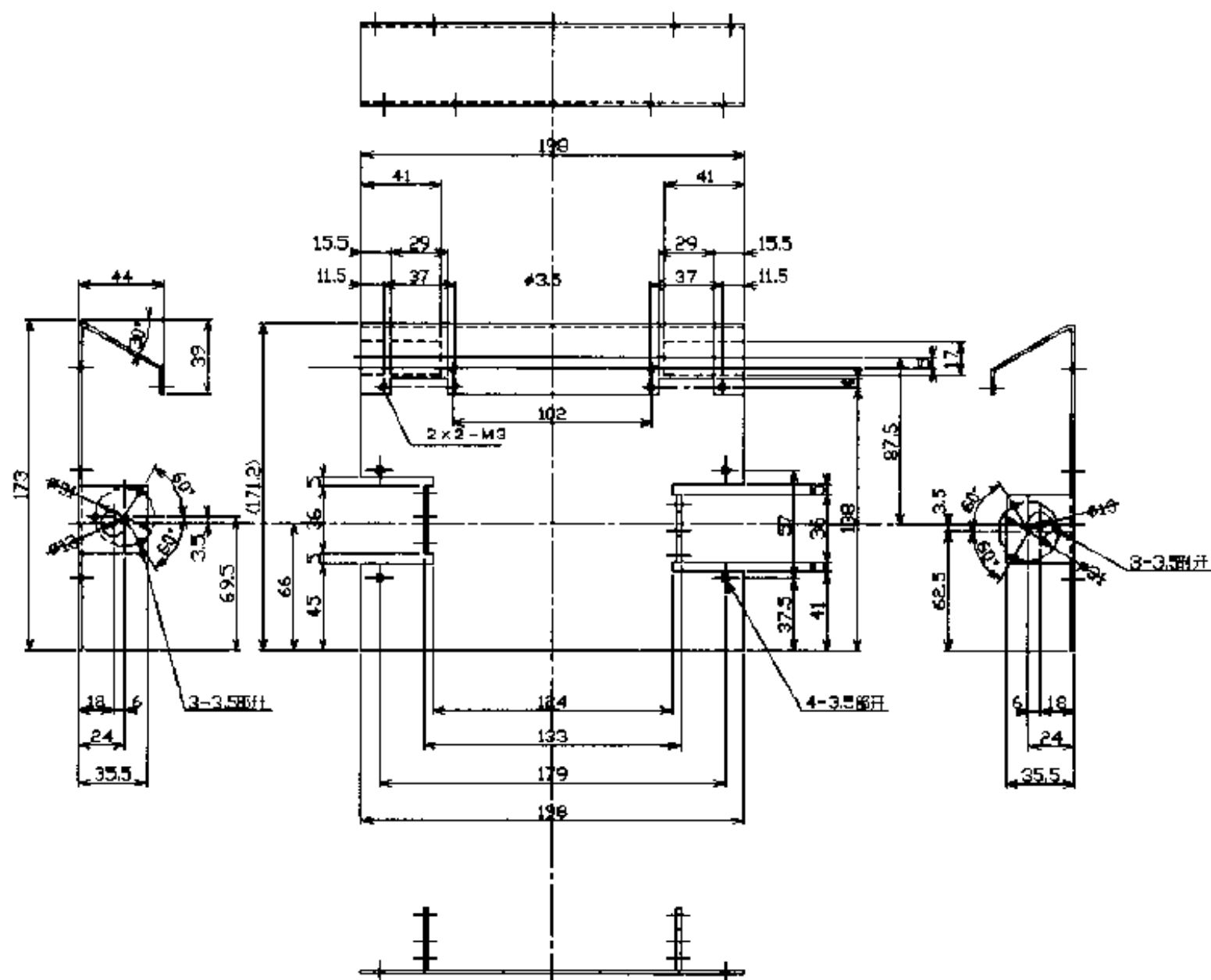
• 照片 10.21 机器人底面



• 照片 10.22 机器人上面

图 10.22 为机器人车身的设计图,用 2mm 的铝板制作。进行吸引的场合,为了防止变形必需使用 2.5~

3mm 左右的厚铝板。电动机安装部分等,对于折弯困难的情况,可使用角铝等制作。



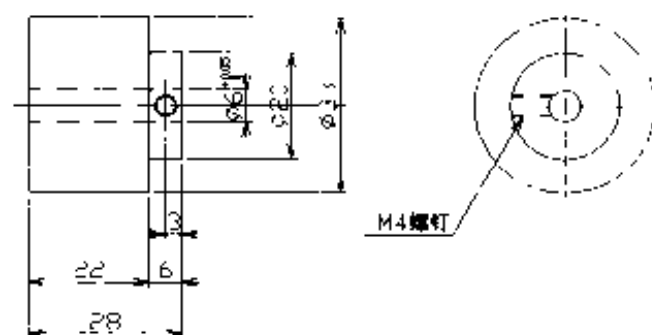
• 图 10.22 机器人车身设计图

10.6.4 轮胎的制作

橡胶轮胎在无线遥控商店有售,这种轮胎适合使用在无线遥控车的前轮。这里使用外径为 45mm 左右的轮胎。

制作适合轮胎的车轮。图 10.23 为车轮的设计图。这是将车轮直接安装在减速齿轮轴上的方式。这种方式

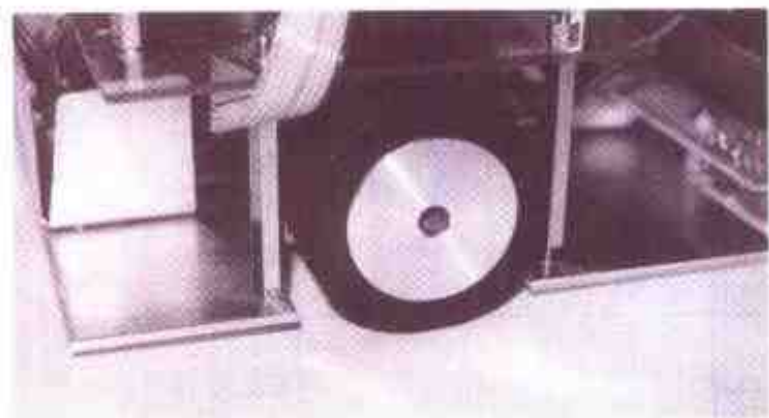
比通过齿轮和皮带驱动车轮的方式制作简单。



• 图 10.23 车轮的设计图

这种车轮制作困难的时候,可通过改造无线方向遥控车用的车轮来制作。到无线遥控商店商谈一下看。有关系密切的无线遥控商店时,在供应产品上方便,所以到有熟人的商店最好。

照片 10.23 是用两面胶带将轮胎粘接在车轮上,然后安装在机器人上。电动机使用“日本伺服”制作的 DME34BA。选择减速比为 18 的齿轮。电动机和减速齿轮组合在一起的形式是 DME34B37G18A。



• 照片 10.23 轮 胎

由这种减速齿轮和电动机的特性及轮胎外径,大体上可以预测机器人的速度。电动机的速度在空载时为 5000r/min,所以带有减速齿轮时的最高转速为 $5000/18 = 278\text{r/min}$ 。将电动机和减速齿轮成套购买时,在其商标上记载 278r/min。如使用的轮胎直径为 45mm,所以它的线速度为: $45\text{mm} \times 3.14 \times 278\text{r/min} = 39281\text{mm/min}$,则每秒线速度为 $39281\text{mm/min} \div 60/1000 = 0.654\text{m/s}$ 。

10.6.5 摔跤场传感器和对手传感器

在检测摔跤场用的传感器中,使用

第 8 章叙述的调制型光传感器 4 个。

对相扑机器人,使用光传感器或超声波传感器作为对手传感器。这里使用能够比较容易得到的光传感器。这种光传感器的原理在前面已经叙述,它带有高辉度的 LED,只要在红外线能到达的距离即可。由于光量多,所以瞬间流过大电流,检测其反射光。距离变远时,由于反射返回来的光量减少,所以需要透镜聚集。

制作这个电路简单,而透镜等较难买到,所以作为反射型光传感器,还是决定使用市场上出售的传感器。

照片 10.24 为由 Keyence 会社出售的反射型光传感器(PZ41)。检测距离为 60cm,另外,还有检测长距离的光传感器。

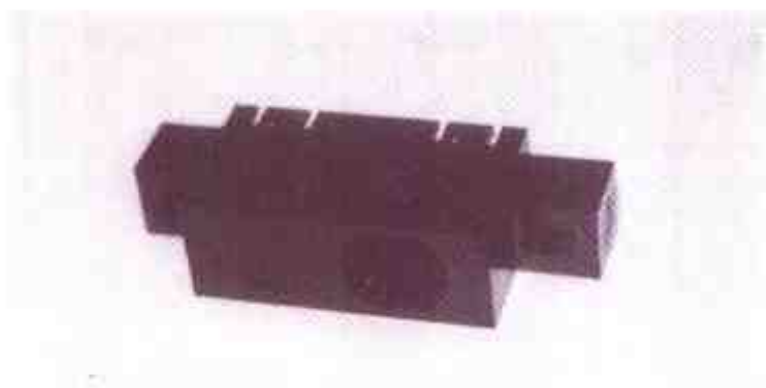
Keyence 的网址如下:

<http://www.Keyence.co.jp/Kouden/index.html>



• 照片 10.24 反射型传感器(PZ41)

照片 10.25 为由最佳技术株式会社 (<http://www.besttechnology.co.jp/>) 出售的反射型光传感器 (BTE023), 这个传感器的检测距离为 80cm。



• 照片 10.25 反射型传感器(BTE023)

最近从价格方面考虑, 决定使用这种传感器。如果连接 5V 电源, 就能得到 ON-OFF 的输出。输出为集电极开路型(晶体管的集电极开路输出), 在这里连接 $1k\Omega$ 左右的电阻, 将另一方连接 5V 电压时, 就能输出。

10.6.6 传感器插件板的制作

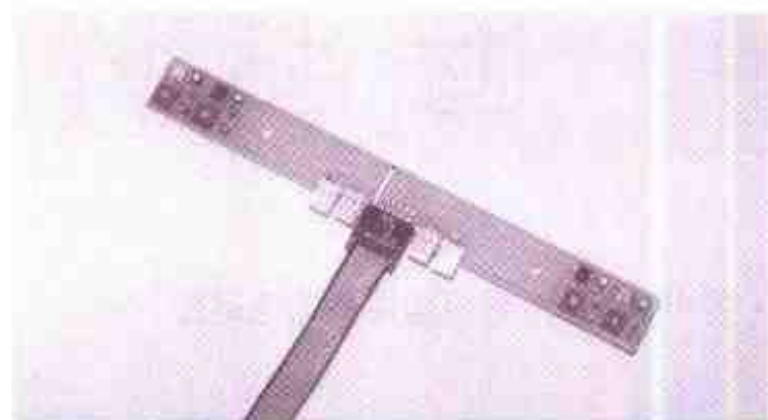
为了实现在 10.6.2 节叙述的附加功能, 摔跤场要安装 4 个传感器。为了让机器人沿着摔跤场行走, 一侧最低需要 2 个传感器。2 个传感器反应的时候, 传感器表明机器人在摔跤场的白线上; 2 个传感器都没有反应时, 表明机器人已离开摔跤场的白线了。这就是所谓一边进行判断, 一边围绕摔跤场旋转。

另外, 设计摔跤场的人将分界线的颜色分为白色、黑色和中间的茶色颜色, 就是考虑希望能够判断分界线。

2 个传感器中的一个可以判断分界线; 调试另一个传感器使其仅能够判断摔跤场的白色线条。2 个传感器, 也许其中的一个仅能够判断白色线条的传感器反应时, 判断摔跤场为白线。能够判断分界线的另一个传感器反应时, 判断分界线。例如, 在分界线位置时, 还没有达摔跤场的界限, 由于还有相当一段距离, 也可以一边加速, 一边检测判断。

图 10.24 为传感器插件板电路图。由摔跤场传感器部分和对手传感器电路两部分组成。

照片 10.26 为制作完成的光传感器插件板。左、右各安装 2 个光传感器。



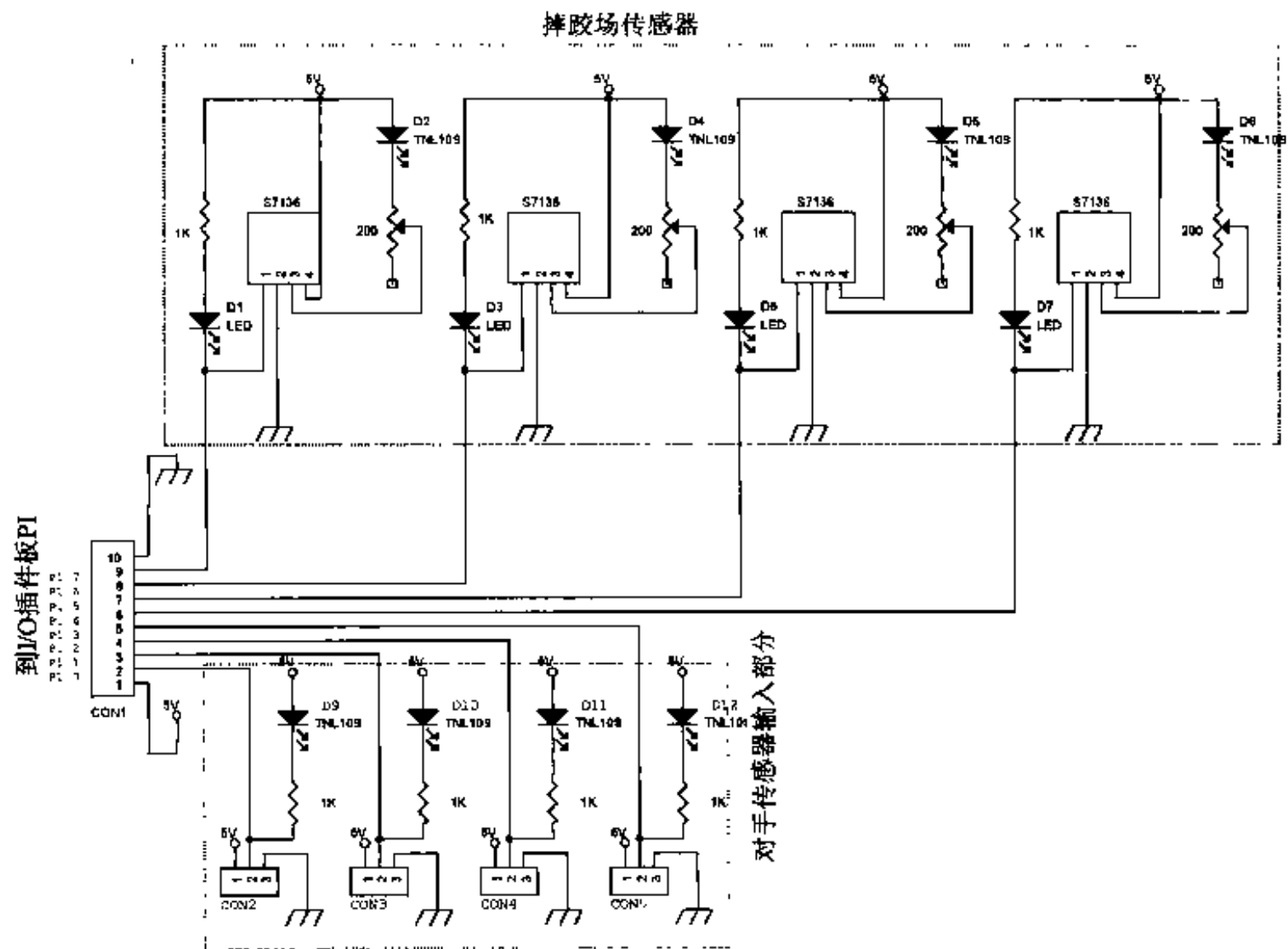
• 照片 10.26 制作完成的传感器插件板

照片 10.27 为传感器部分的放大图形。这次在外侧和内侧安装了具有不同感度特性的传感器。在两侧都可以使用 S7143。引脚的配置相同。

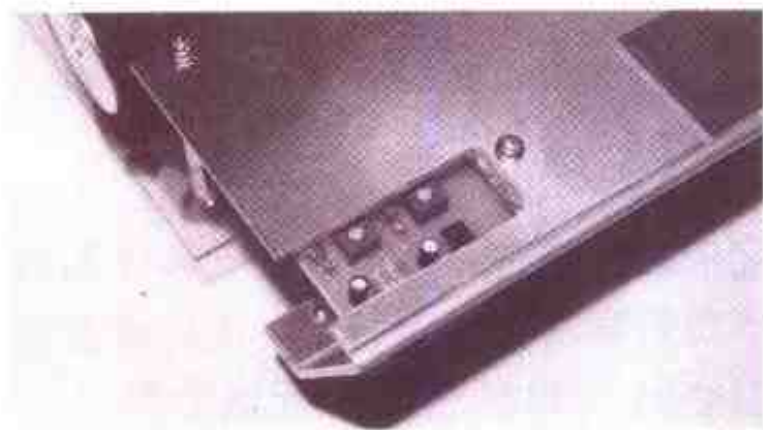
对于红外 LED, 用热收缩管包裹侧面, 以使红外线不能直接射入传感器。用可变电阻调整检测感度。这个电阻可能是安装摔跤场表面, 也可能安装在摔跤场的背面。

照片 10.28 为将这个传感器插件

板刚刚装载在机器人上的情况,连接在 H8 插件板的端口 P1 上。



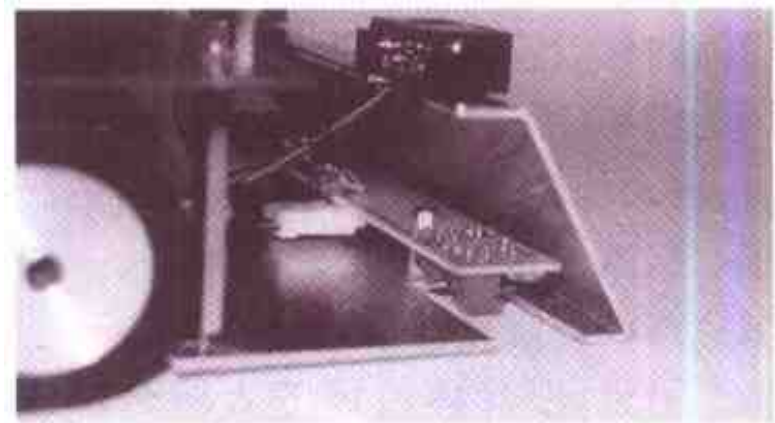
• 图 10.24 传感器插件板的电路图



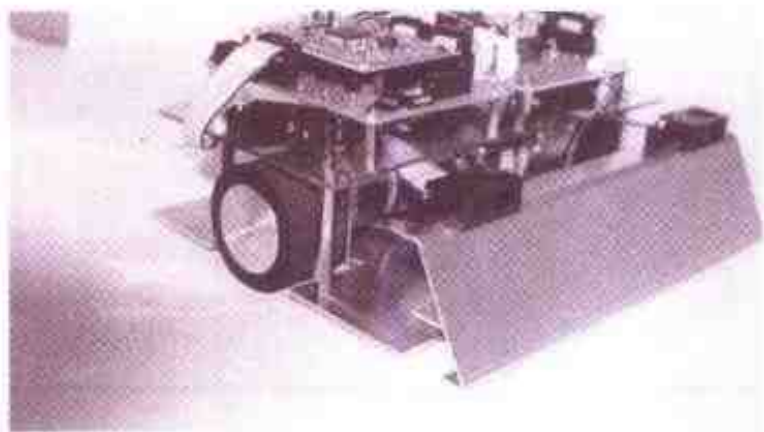
• 照片 10.27 制作完成的传感器插件板的传感器部分

照片 10.29 是将对手传感器刚刚安装在机器人上的情况。中间的 2 个传感器朝向前面,左、右的传感器各向

右和向左倾斜。由此感知对手,进行追踪。左右传感器大体上倾斜 45° 安装。



• 照片 10.28 传感器插件板安装在机器人上



• 照片 10.29 对手传感器安装状况

1. 相扑机器人的程序

CPU 各出口与电动机及传感器的连接如表 10.4 所示。

作为机器人的基本性能是不能从摔跤场跌落,而且能够追踪对手,要将这些部分编制成程序。

表 10.4 CPU 和传感器、电动机的连接

位	P1 的设定	P2	P6	PA	PB
0	摔跤场左外光传感器	LED 位 0	数字开关位 0	旋转编码器输入	备用电动机 1PWM
1	摔跤场左中光传感器	LED 位 1	数字开关位 1	旋转编码器输入	备用电动机 1ON-OFF
2	摔跤场右中光传感器	LED 位 2	数字开关位 2	右电动机 PWM	备用电动机 2PWM
3	摔跤场右外光传感器	LED 位 3	数字开关位 3	右电动机 ON-OFF	备用电动机 2ON-OFF
4	对手左倾斜光传感器	LED 位 4	按钮 /SW	左电动机 PWM	—
5	对手左前光传感器	LED 位 5	—	左电动机 ON-OFF	—
6	对手右前光传感器	LED 位 6	—	—	—
7	对手右倾斜光传感器	LED 位 7	—	—	—

图 10.25 所示的是这次制作的相扑机器人程序的流程图,这里将重点进行讲解。

2. CPU 的初始化

硬件的连接如表 10.4 所示,所以这里必需进行使用端口的初始化。这些地址的定义如表 10.5 所示。

为将输入位置 0、输出位置 1,写入端口数据定向寄存器地址即可。这时的地址只要按照标题定义即可。例如,P1DDR 如果写为 P1.DDR,用字节单位可以存取。



• 图 10.25 相扑机器人程序流程图

表 10.5 各端口的地址及其定义

端口	端口数据定向寄存器地址	简称	端口数据寄存器地址	简称
P1	H'FFC0	P1DDR	H'FFC2	P1DR
P2	H'FFC1	P2DDR	H'FFC3	P2DR
P6	H'FFC8	P6DDR	H'FFCB	P6DR
PA	H'FFD1	PADDR	H'FFD3	PADR
PB	H'FFD4	PBDDR	H'FFD6	PBDR

3. 开场模式

用数字开关 16 可以设定开场模式。

这个开场模式必需对各种各样的对手有所准备。特别是对对手具有臂结构的机器人,这个开场模式变得更重要。

最初通过选择数字开关值选定开场模式并开场:

4. 竞赛的模式

开场模式一结束,就进入竞赛模式。竞赛模式要检测,追踪对手。另外,如果检测摔跤场,要后退,改变方向。这时,对应左、右的摔跤场传感器值,改变旋转方向或旋转时间。

摔跤场传感器检测后的处理如图 10.26 所示。例如,右边摔跤场传感器反应时,机器人要后退,向左旋转。若向右旋转则旋转时间要长,将背面向对手的时间变长。

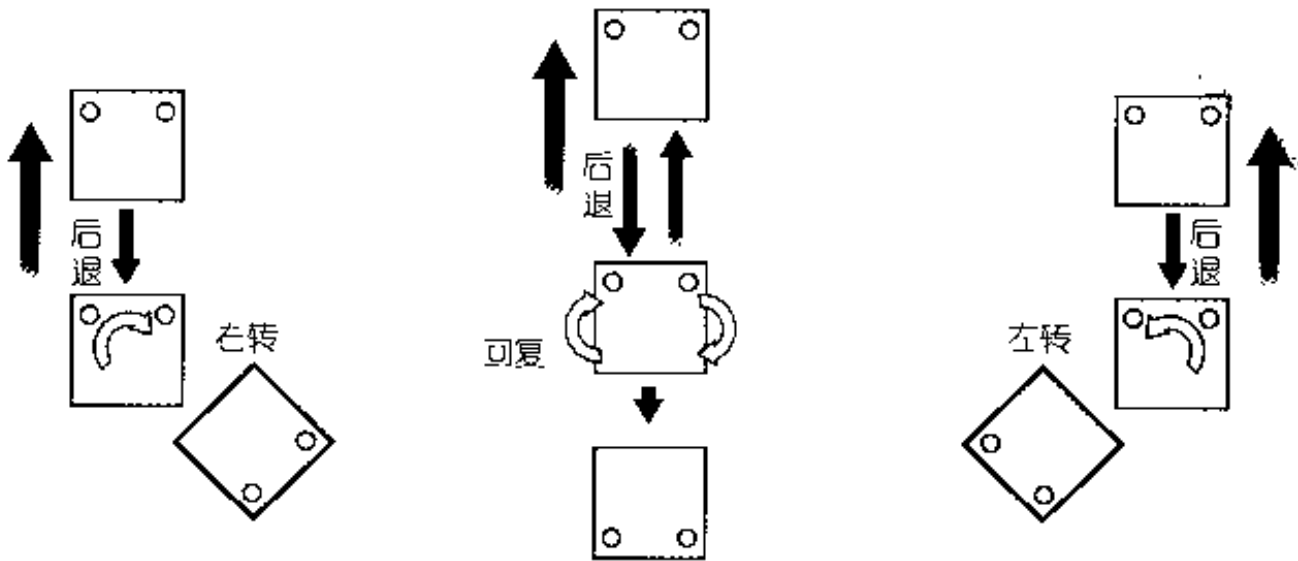
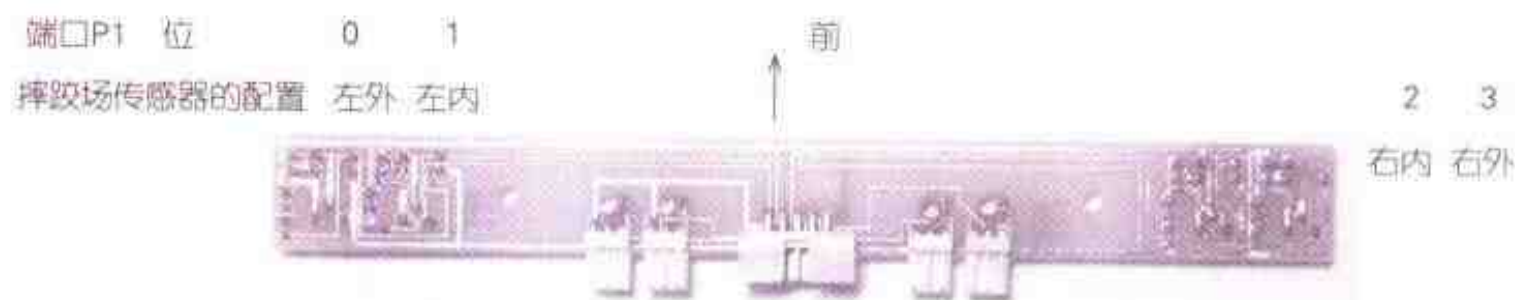


图 10.26 察觉摔跤场传感器后的处理

摔跤场传感器的配置及对应输入端口的位(bit)在照片 10.30 中表示。这次在左、右分别安装 2 个摔跤场传

感器,所以如表 10.6 所示,有 16 种反应模式。用这个传感器输入例行程序判断读入的数据,决定处理。



• 照片 10.30 摔跤场传感器的配置和对应输入端口位

表 10.6 摔跤场检测后的处理

10 进制数	2 进制数	检测后的处理
0	0000	前进
1	0001	右转
2	0010	后退、右转
3	0011	后退、右转
4	0100	后退、左转
5	0101	后退、左大转
6	0110	后退、左大转
7	0111	后退、左转
8	1000	左转
9	1001	后退、左大转
10	1010	后退、左大转
11	1011	后退、左大转
12	1100	后退、左转
13	1101	后退、左大转
14	1110	后退、左大转
15	1111	后退、左大转

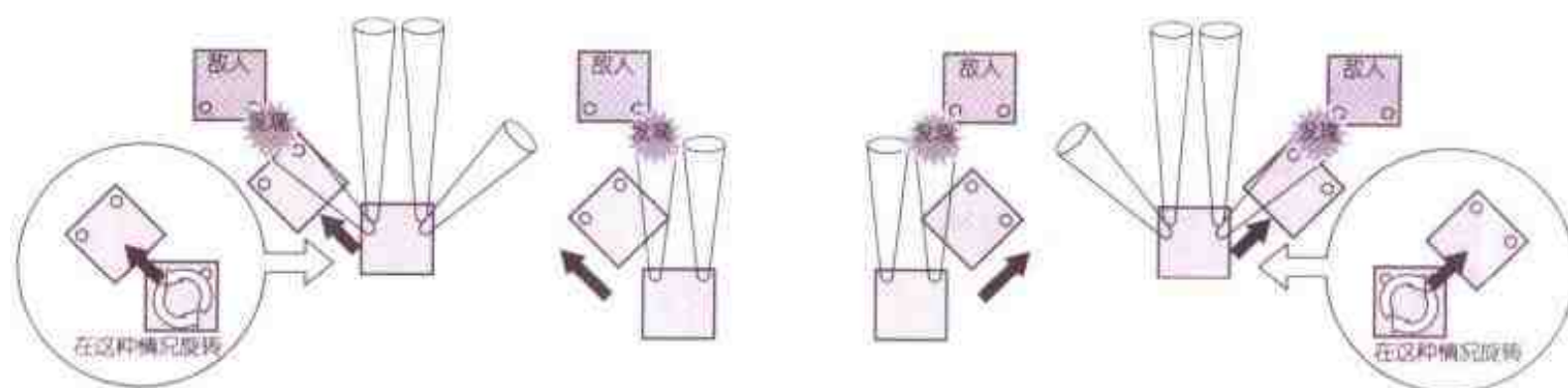
摔跤场传感器的反应 ○○ ○○(0000), 所有的传感器没有察觉摔跤场。因而, 这时要检验对手传感器, 输入追踪对手程序。其中任

一个闪烁时, 都可进行摔跤场的界限的处理。

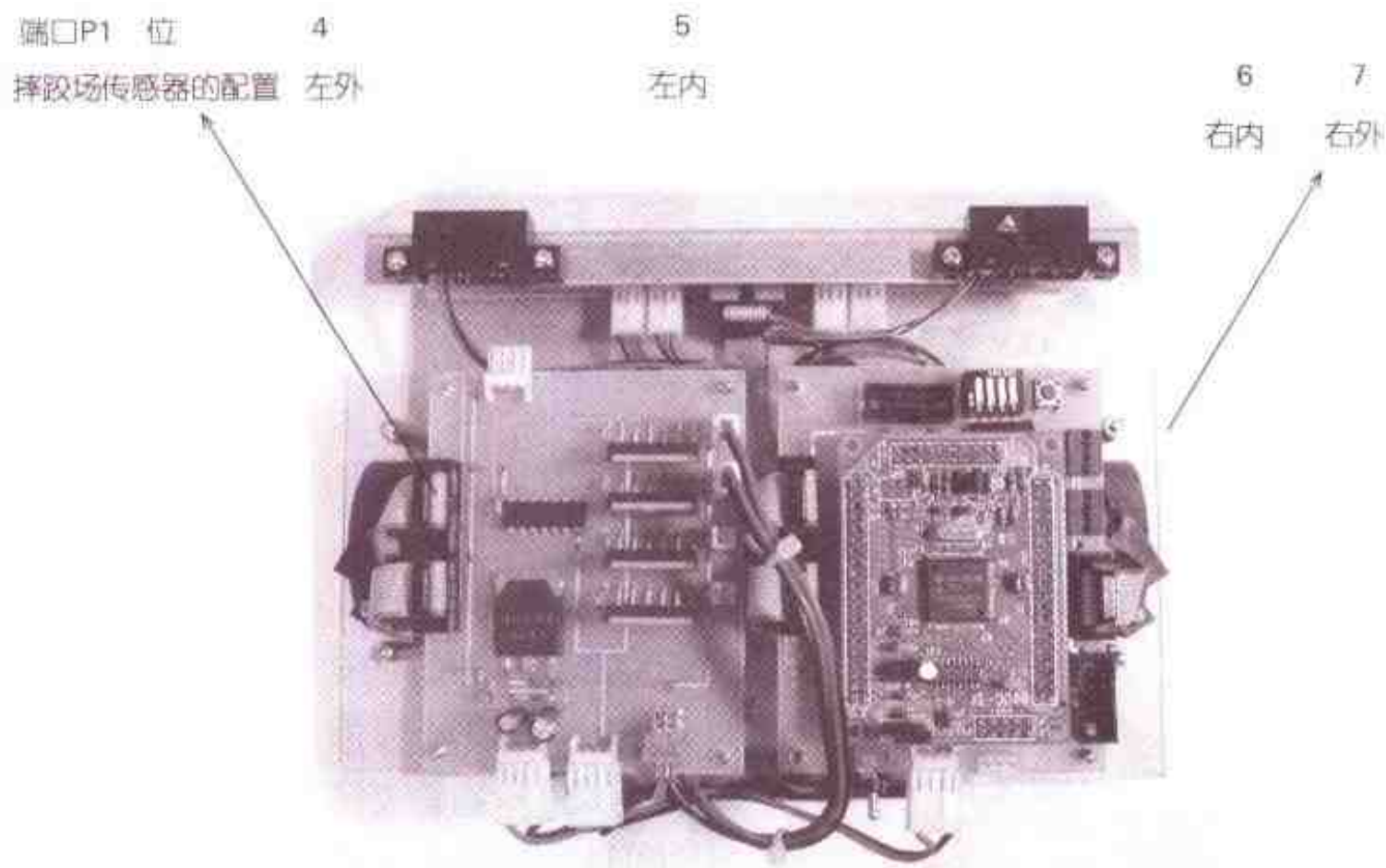
例如, 如果变成 ○○ ○ ● (0001), 右外侧的传感器闪烁, 所以右边进入摔跤场界限。这时, 稍微向左转弯看一看。如果变成 ○○ ● ● (0011), 右传感器两方闪烁, 所以后退向左转, 返回竞赛模式, 就是这样, 传感器检测后再做处理, 防止机器人从摔跤场上跌落。

对手传感器检测后的处理按图 10.27 所示进行。只要对手在右侧时向右转弯; 在左侧时向左转弯即可。

如果检测到对手如何处理呢? 和检测摔跤场相同, 只要根据各自的输入分析即可。照片 10.31 示出了与对手传感器的配置相对应的输入端口的位。



• 图 10.27 对手传感器检测后的处理



• 照片 10.31 对对手传感器的配置输入端口位

对手传感器的状态如果变成●○○○(1000),对手在右侧。因而,只要进行右大转弯即可;如果变成○●○○(0100),可知仅稍微向右转弯即可。如果变成○●●○(0110),则一直前进。变成●○○●(1001)时,也许稍微觉得头疼,而对手如果是扩展臂型的机器人有可能在前面,因而前进。参考图 10.27,对其他情况进行同样分析即可。将有关全部对手传感器的模式归纳于表 10.7。

5. 机器人调整程序

在个人电脑的画面上可以调整机器人的动作,下面介绍简单的机器人调整程序。

作为调整对象的机器人的动作如下:

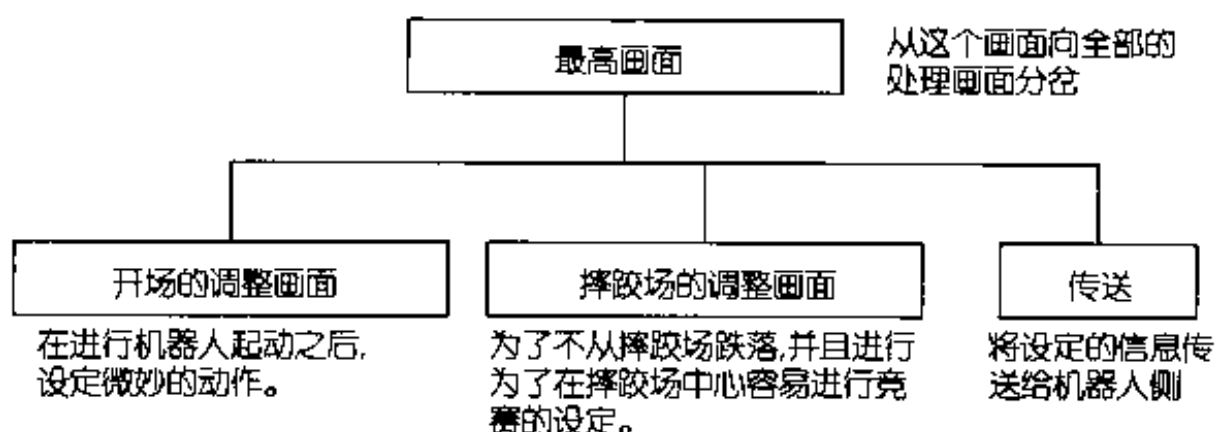
- 开场动作;

- 摔跤场的界限(外侧)跌落回避动作;
- 电动机的旋转方向。

表 10.7 对手检测和处理

10 进制数	2 进制数	处 理
0	0000	前进
1	0001	左回转
2	0010	左转
3	0011	左转
4	0100	右转
5	0101	前进
6	0110	前进
7	0111	前进
8	1000	右回转
9	1001	前进
10	1010	左转
11	1011	前进
12	1100	右转
13	1101	前进
14	1110	前进
15	1111	前进

基本的画面构成如图 10.28 所示,画面 10.1 为开场的调整画面。



• 图 10.28 基本的画面构成

6. 使用臂实践的程序

对于上述的基本程序,进行打倒臂、一动不动地保持部分的补加和追踪对手程序的变更。有关硬件的变更部分在后面讲解。

7. 面向对手方向控制

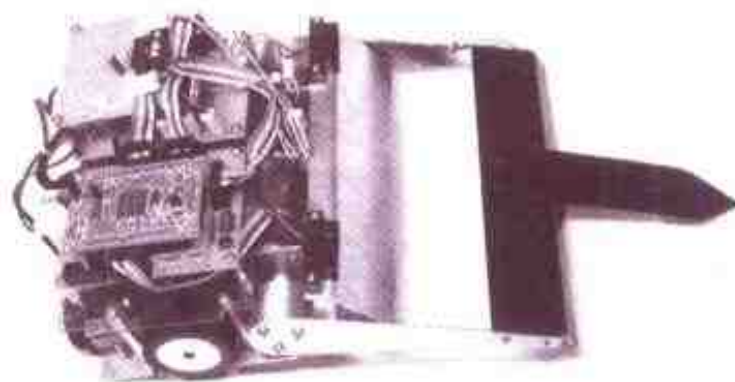
如果经常攻击对手的正面,对手一定开到机器人臂上。这时,臂的顶端要锋利地加工好。事先将顶端作尖,打倒臂的状态对摔跤场稍有损伤。以这种状态使机器人前进时,将损坏摔跤场。因而,这里,这个尖臂的顶端仅仅是在中心旋转。在这里如果一动不动地等待,当然是进攻对手的正面。推对手时,一定抬起臂。这时,要用臂的顶端,但又要设法决不能损坏摔跤场。

还有,臂的表面用黑色涂敷消除光泽(参见照片 10.32),以使对手不易觉察已被推到了赛场边界,从而逃脱。

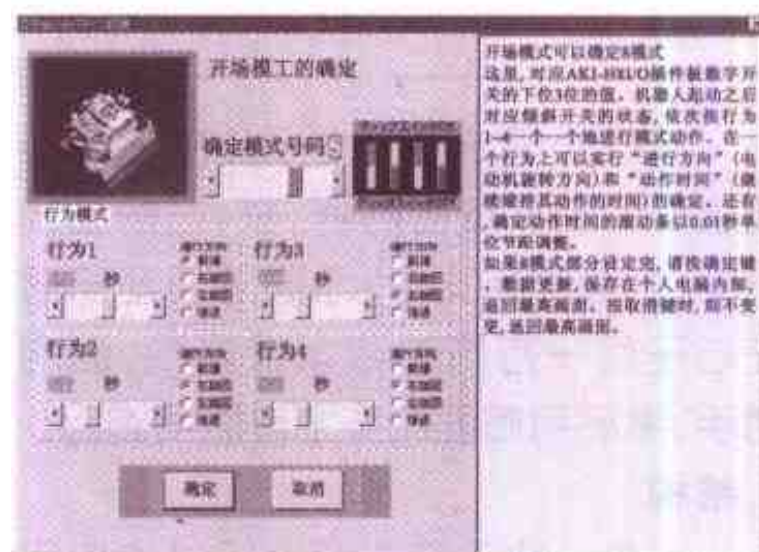
8. 带臂机器人的改造

最近吸引型机器人多是力量型相扑机器人,对观赏者来说,有时感到没有意思。给机器人安上臂,等待对手

开到臂上,切断吸引再将其推出,对机器人进行这样的改造。



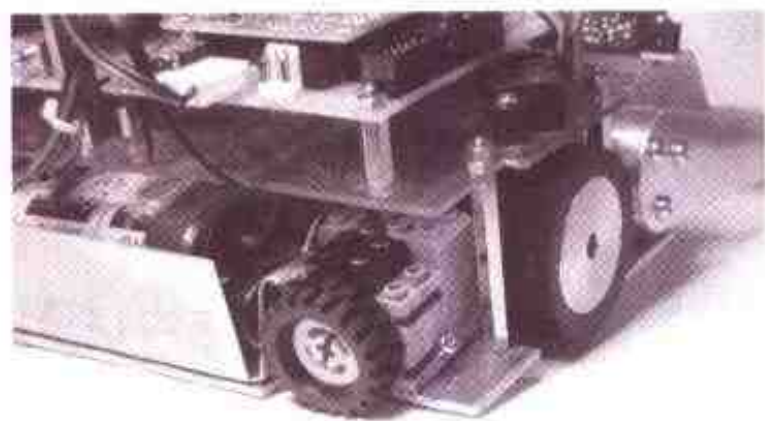
• 照片 10.32 臂的顶端



• 画面 10.1 开场模式调整的确定

经常面向对手的控制如照片

10.33所示,在其后面安装电动机。这里使用 LEGO 的电动机,而模型是用 MABUCHI 电动机。使用伞型齿轮或涡轮,在直角上安装驱动轴,装载空间能充分保证。将两个车轮安装在后面,而其中一个是假轮,只有轮胎。



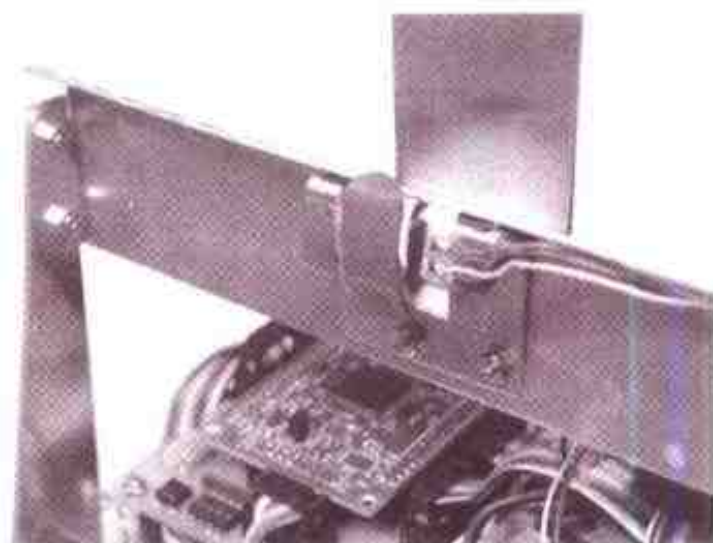
• 照片 10.33 机器人后部的车轮

通过对手传感器控制这个电动机。如果对手进入左侧就向右运动,进入右侧就向左运动,用这样简单的程序,就完全可以进攻对手正面。

9. 感觉对手重量的传感器

利用第 8 章讲解的感压传感器判断对手开到臂上的情况。如照片 10.34 所示,机器人臂用双面胶带黏接,用电话卡导向。而且,为了容易滑动贴上家具滑板。

与微型计算机的连接,是将臂用电位器连接在 AN0 上,将传感器连接在 AN1 上;将 PB0 连接在臂用电动机 PWM 输出上。



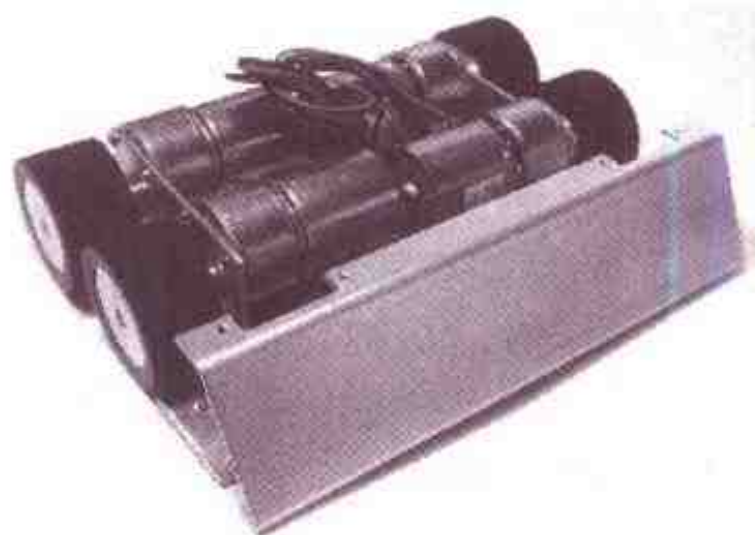
• 照片 10.34 安装感压传感器

10.7 高速行驶的机器人

本节介绍高速机器人,在它前面安装 2 个超声波传感器,左、右各安装一个、后面安装 2 个,无论对手在前后左右任何一方向都可以探测到,面向对手,然后用前面的 2 个传感器追踪。

1. 结构

这是一种在第 9 章介绍过的使用 8 通道超声波传感器的相扑机器人。这种机器人旋转脚如照片 10.35 所示,用 4 轮驱动。

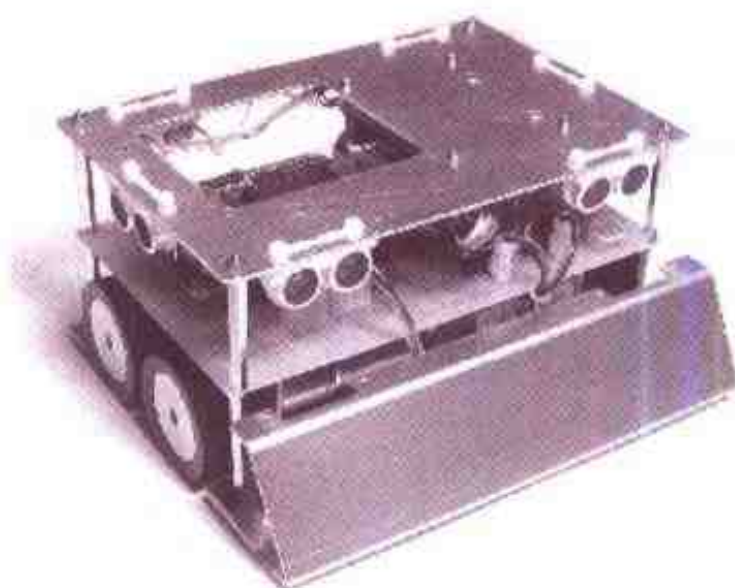


• 照片 10.35 旋转脚

照片 10.36 是机器人外型。在机器人平面上全部安装好超声波传感器,它可以 360° 捕抓对手。传感器在机器人前面、后面各 2 个,左右各 1 个,一共 6 个传感器。这个传感器通过可变电阻调节,可以检测 1m 以上的距离。

2. 带有 6 通道超声波传感器机器人的程序

它是一种检测到摔跤场就进行 U 型回转,发现对手则追踪的程序。

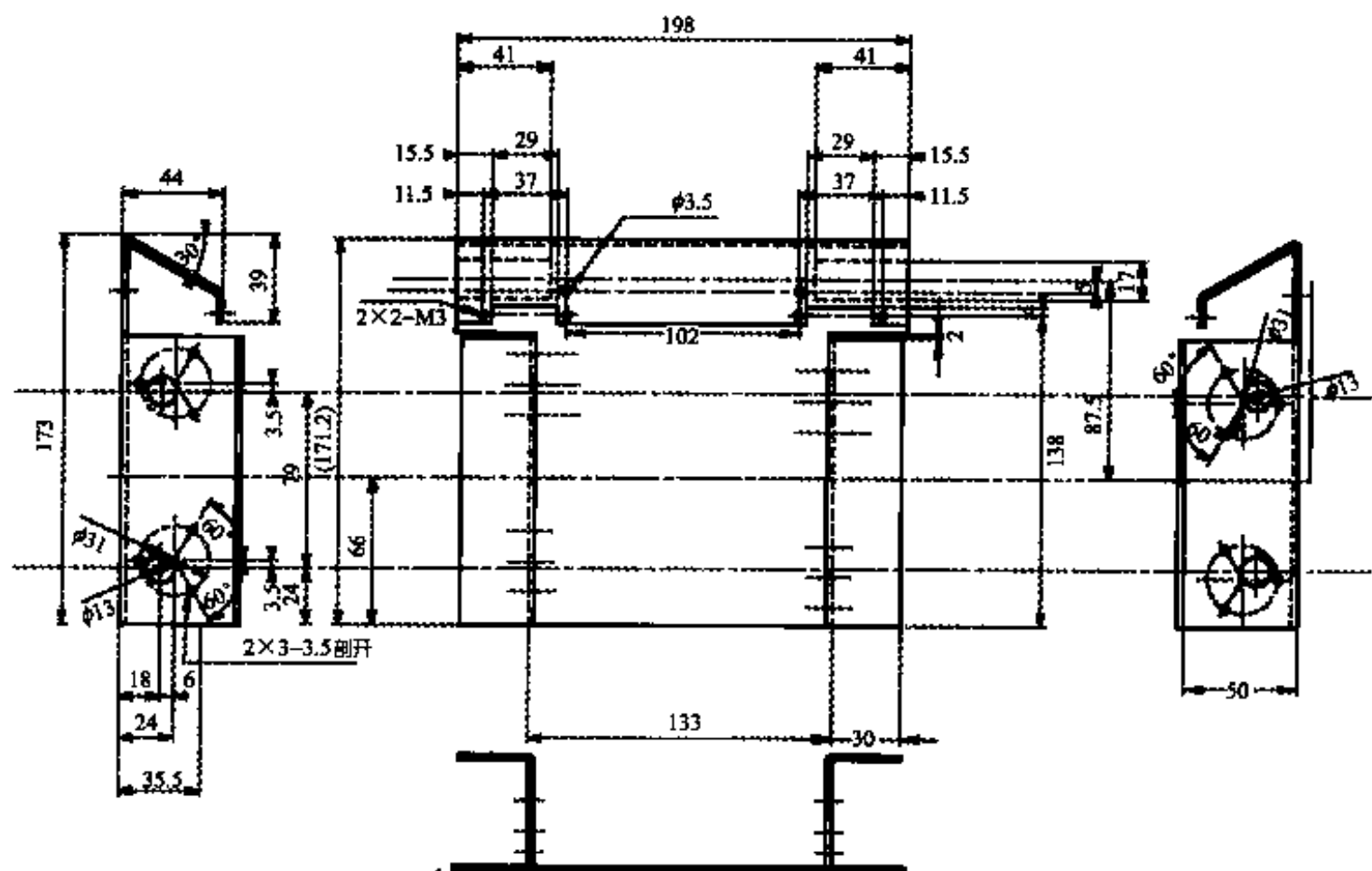


• 照片 10.36 高速机器人外型

10.8 吸引型机器人的制作实例

本节介绍吸引型机器人的制作。车身以 4 轮驱动,电动机使用前面叙述的日本伺服电动机。其中,这里采

用大减速比为 1/61。要求具有克服吸引的能力,作为参考,车身设计图如图 10.29 所示。



• 图 10.29 车身的设计图

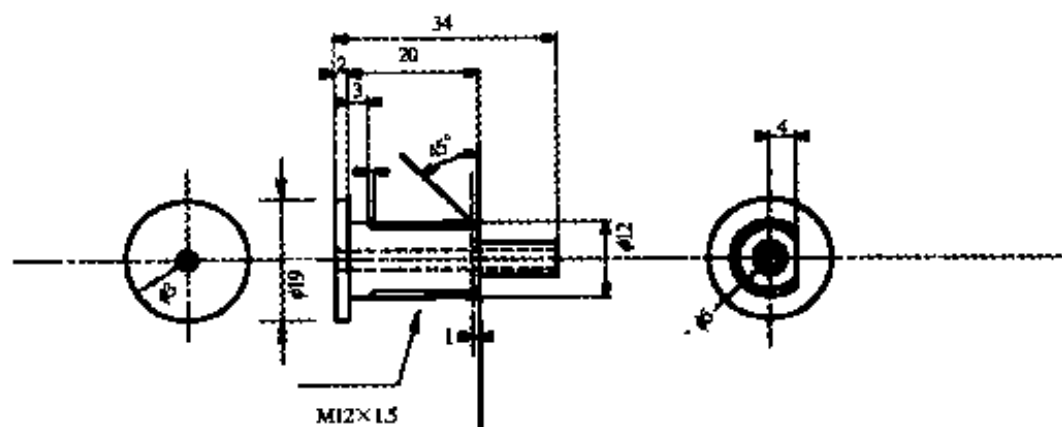
1. 吸盘的固定

吸引泵使用前面介绍的 Mede 产业的 DP0140, 吸盘也使用前面介绍过的产品。固定吸盘, 需要制作图 10.30 所示的螺栓, 用它将吸盘和车身的基板夹住固定。照片 10.38 是刚刚固定

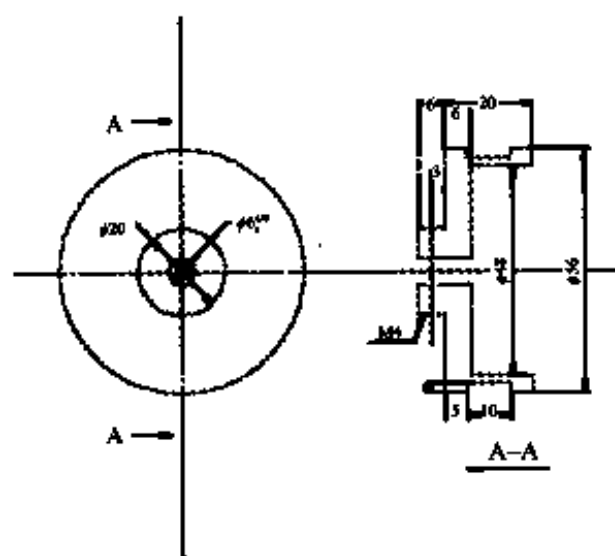
的吸盘。

2. 车轮

带轮机器人, 开始吸引的时候, 为了防止下沉, 车轮几乎全部用铝车轮, 在其轮上卷缠橡胶薄片。图 10.31 是车轮设计图。



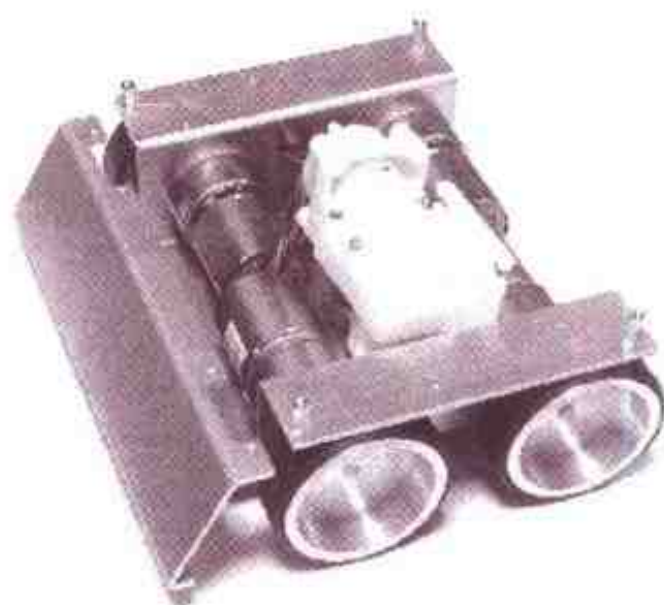
• 图 10.30 吸盘安装螺栓



• 图 10.31 车轮的设计图

由此, 与利用橡胶车轮相比, 由于开始吸引时引起车轮下沉, 可以抑制“吸盘过分吸住”。

另外, 在吸盘的下面黏接聚四氟乙烯薄片, 可以降低摩擦阻力, 使之平滑地运动。



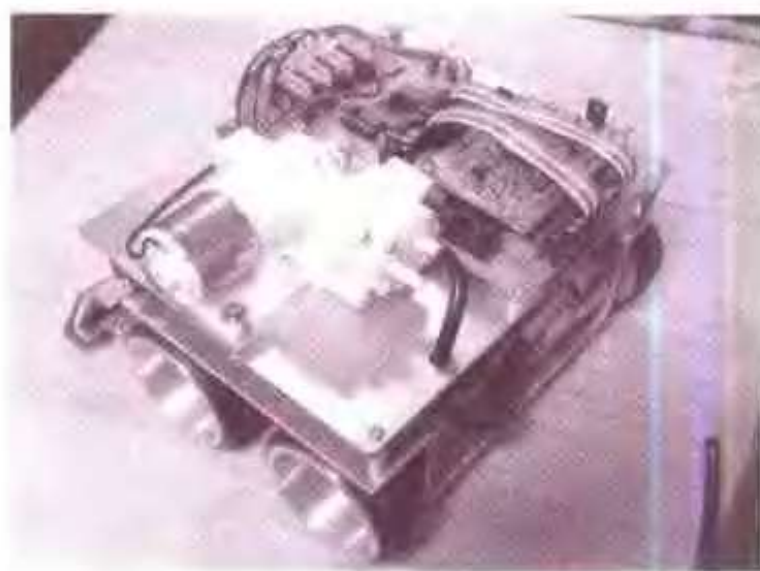
• 照片 10.37 车身和泵的配置

程序

它是一种通过 2 个对手传感器, 一边吸引, 一边追踪对手的基本程序。



• 照片 10.38 吸盘的安装



• 照片 10.39 制作完成的吸引机器人

chapter

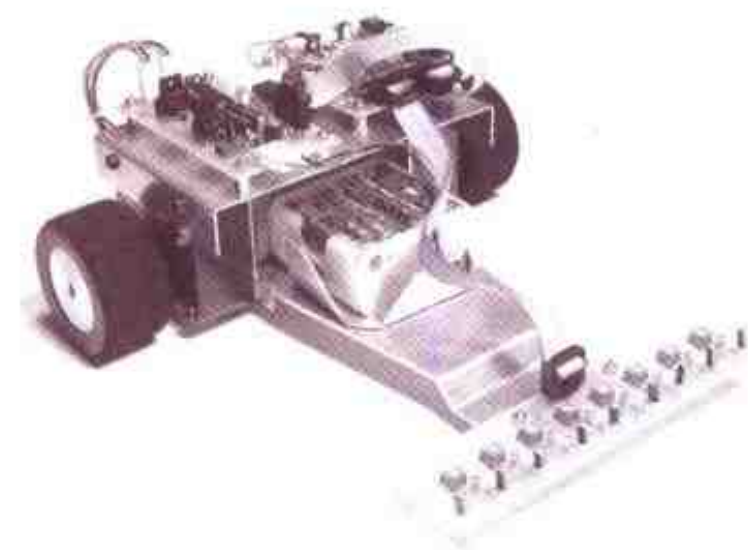
11

直线追踪机器人的 制作方法

本章将介绍在日本微型计算机圣杯机器人或微型鼠大会上进行的机器人竞赛，试制作能够参加机械学会的三叉戟机器人等竞赛的直线追踪机器人（参见照片11.1所示）。制成的机器人分为基本驱动部分和安装直线追踪用的光传感器部分，而且作为其发展型号，也介绍装载转向转换装置或长枪部件的机器人。

11.1 车身的制作方法

这里稍微认识一下日本微型计算机圣杯机器人。对日本微型计算机圣杯机器人使用碱电池已经变成规则,不能使用大电流的电动机。由于碱电池内部电阻大,是造成电源电压降低的原因。为此,引起 CPU 清 0 等等问题。另外,碱电池的容量比镍镉电池的容量小,所以在全路线行走期间,由于电压降低也有问题。因此,电动机的选定很重要,这个电动机应该配合作成的车身。



• 照片 11.1 制作完的直线追踪机器人

1. 电动机的选定

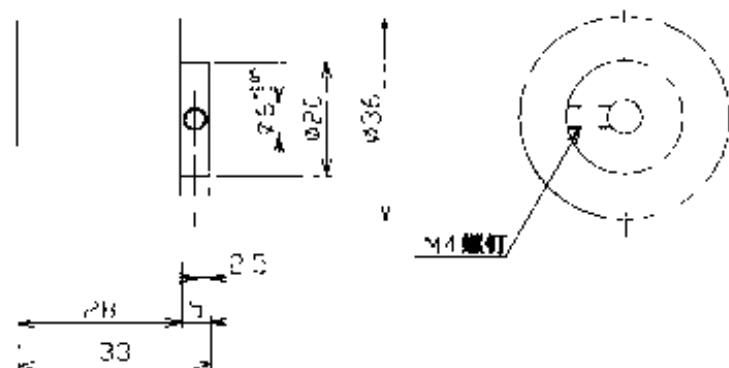
对于电动机使用日本伺服会社制作的 DME34B6HPA,另外,传动头使用 6DG7.5。额定电流为 0.76A,设定使用碱电池的极限值。如果使用镍镉电池,能发挥电池的足够能量。

而且,对日本微型计算机圣杯机器人有直角的拐角,刹车特性也很重

要。通常如果选定额定电压高的电动机,它的电抗大,对刹车有利。这里正是发挥电抗值大的电动机的作用。

2. 轮胎的选定

从选定电动机转速和轮胎直径,可以简单地计算机器人车的速度。但是,有关路面的阻力或上坡路的驱动转矩等成为重要的因素。考虑这点,在轮胎上使用无线方向遥控车的橡胶状产品,车轮尺寸如图 11.1,自己制作。另外,也可以使用无线遥控车用塑胶车轮。

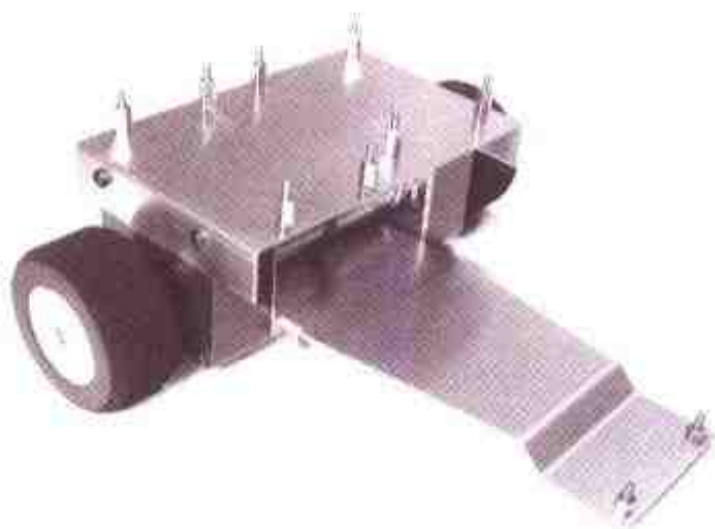


• 图 11.1 车轮设计实例

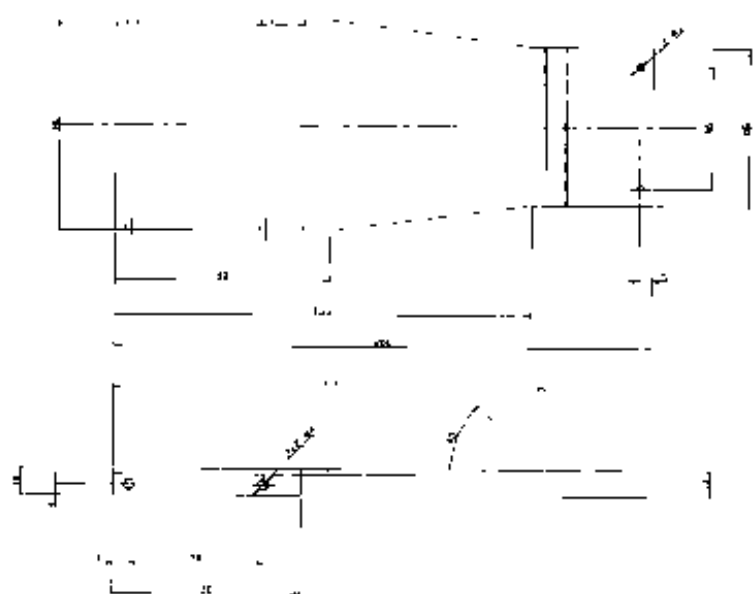
3. 直线追踪机器人的车体

车体按照将电动机配置在后部,尽量将光传感器配置在前面设计即可。光传感器在前面可以很快地获得线路上的信息,利用这个信息控制,所以控制滞后量小。

照片 11.2 为机器人车身,图 11.2 为车身设计实例。



• 照片 11.2 基本的车身



• 图 11.2 车身的设计实例

4. 旋转编码器(光电编码器)

安装旋转编码器时,可以测量速度和距离。对于直线追踪有指示器,离这里前进多少米有拐角,就这样给你一个信息。熟练地利用这个信息时,机器人可以不怎么减速就可以进入拐角。

另外,通过知道速度,在上坡道等速度降低时,使电动机中流过大电流,这样使转矩上升,可以保持电动机速度恒定。于是,通过安装旋转编码器,可以进一步提高机器人的性能。

照片 11.3 为传感器和切口圆盘

各为一体的旋转编码器。由于输出变为方波,所以只连接电源就可以将输出写入微型计算机。



• 照片 11.3 旋转编码器

照片 11.4 为车轮和旋转编码器组装成一体的情况。将其安装在机器人车体的后部,测量距离和速度。



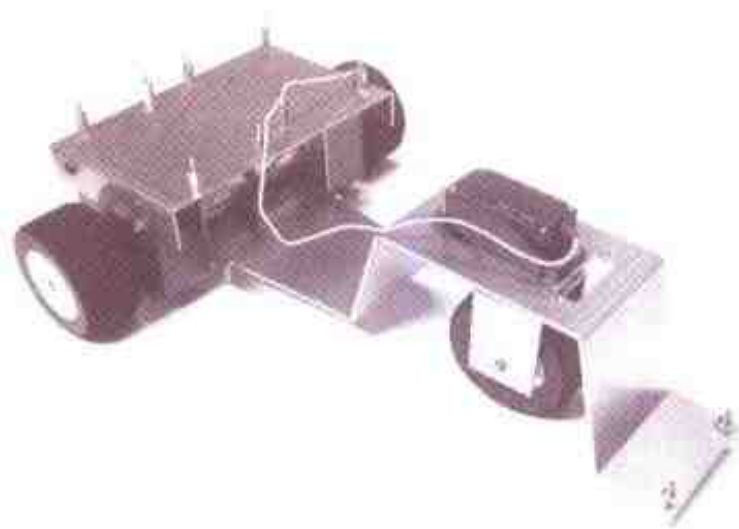
• 照片 11.4 车轮与旋转编码器组装一体

5. 安装转向转换装置的车身

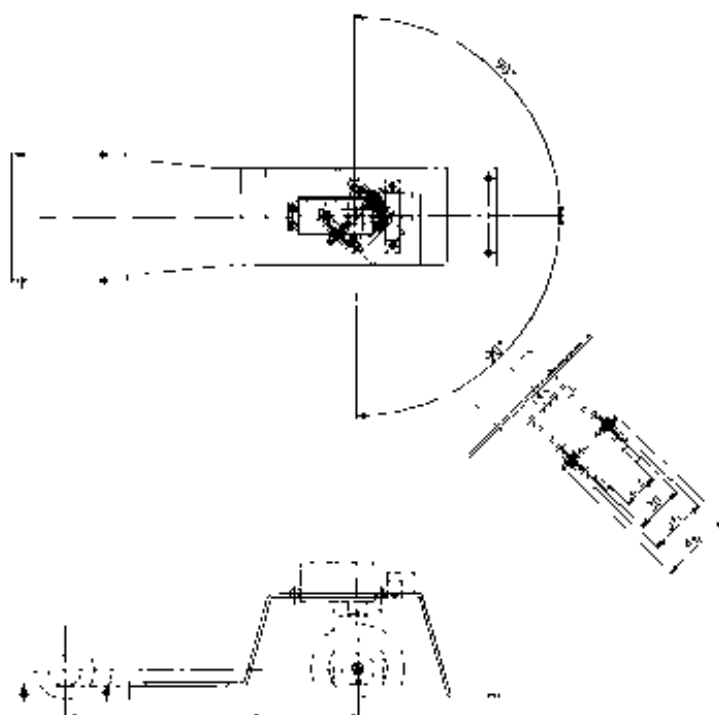
安装转向转换装置时,其行走性能能进一步提高。使左、右电动机转速变化,对其进行控制,由于用转向转换装置进一步控制,所以稳定性进一步提高。

对转向转换装置控制采用无线方向遥控用伺服会社的产品,这里对准

安装车轮,如照片 11.5 所示。另外,其设计图在图 11.3 中表示。



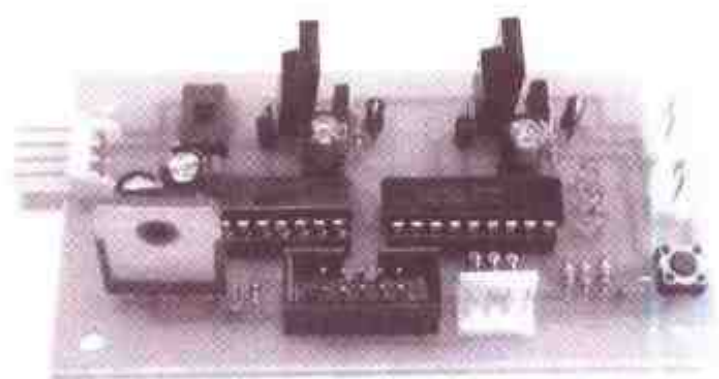
• 照片 11.5 安装转向的车身



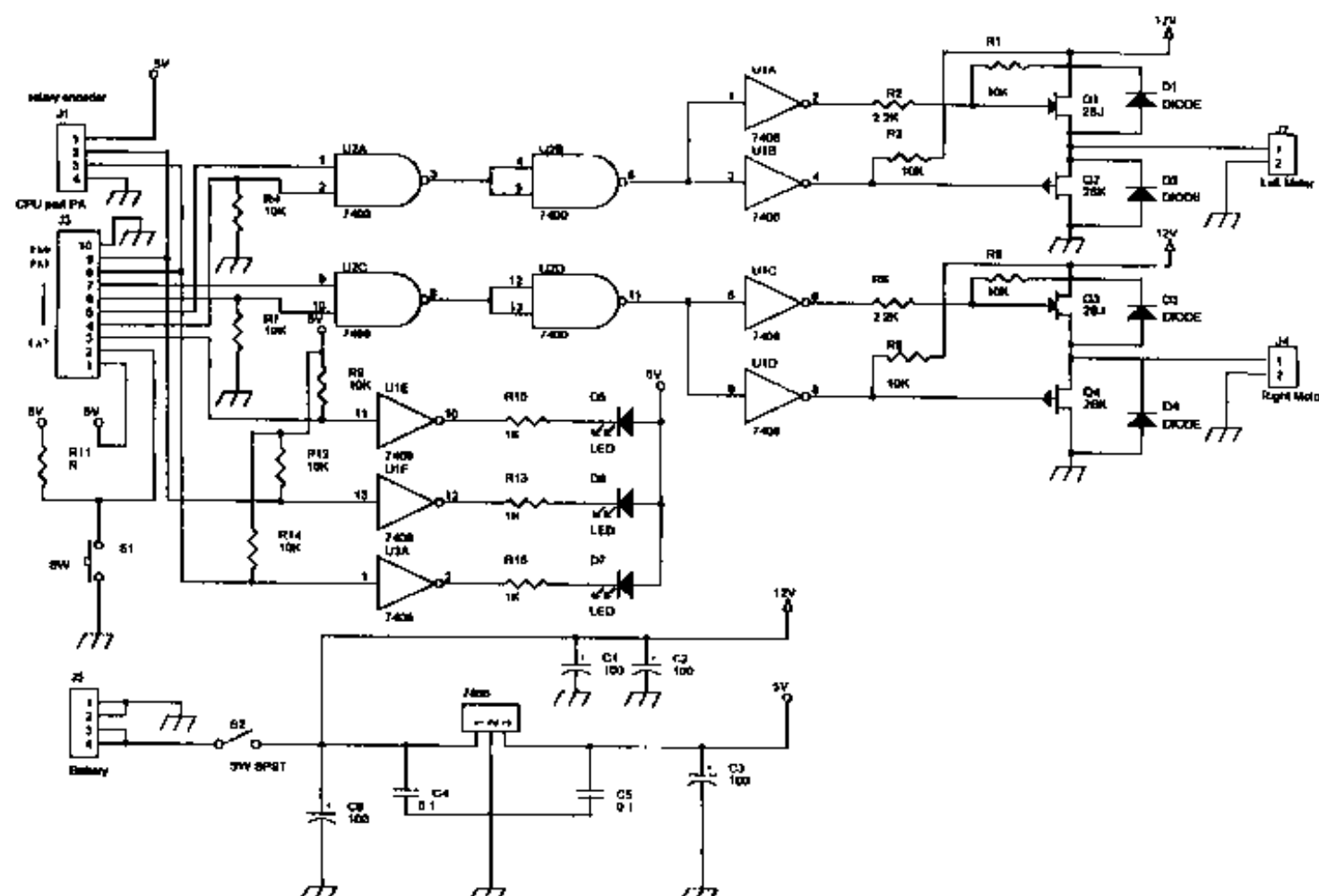
• 图 11.3 安装转向转换装置车身的设计实例

11.2 驱动电动机插件板

图 11.4 为驱动电动机插件板电路图。直线追踪机器人只前进,不需要返回,所以用 2 个 FET,仅将电动机正转和制动作为控制回路。另外,5V 电源电路也内装在插件板上。照片 11.6 为制作完的驱动电动机插件板,而用比较简单的电路构成可以实现直线追踪。



• 照片 11.6 驱动电动机插件板



• 图 11.4 驱动电动机插件板电路图

11.3 光传感器插件板

光传感器插件板就是将第 8 章的调制形光传感器 8 个安装在一起的元件。使传感器间隔在 10mm 以下即可（参见照片 11.7 所示）。



• 照片 11.7 光传感器插件板

11.4 安装长枪部件

对于参加三叉戟机器人竞赛，为了撞击到目标，必需安装长枪部件。这里使用带有转向转换装置的车身，在转向转换装置用伺服机构上，安装上和其完全相同的枪用部件。加工比

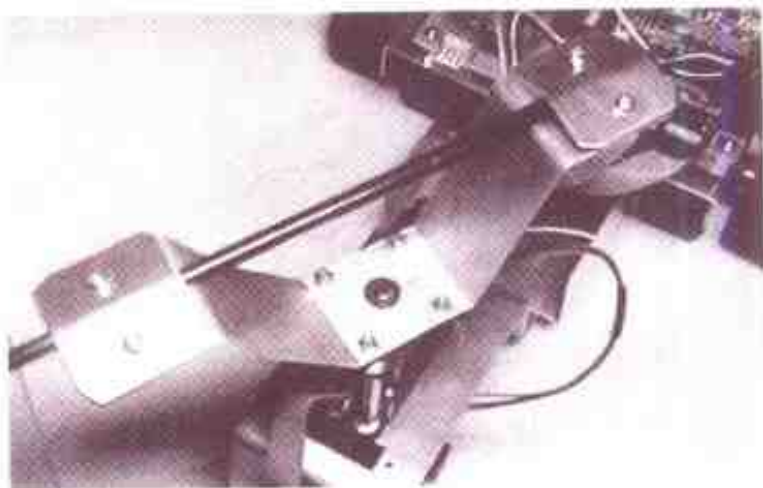
较麻烦，与转向转换装置的伺服机构安装部分共享，用间隔固定（参见照片 11.8 所示）。

在铝板上安装长枪部件，如照片 11.9 所示。在长枪部件上使用钓鱼竿

顶端,其长度离伺服机构中心 400mm。



• 照片 11.8 三叉戟机器人

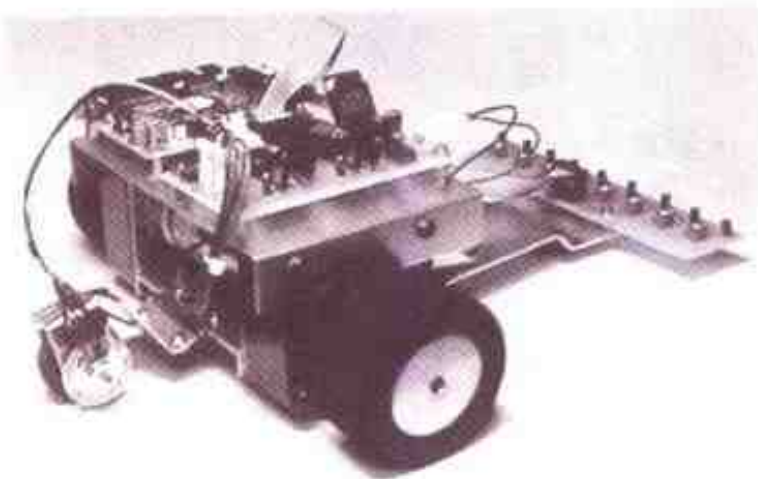


• 照片 11.9 安装长枪部件

11.5 机器人的程序实例

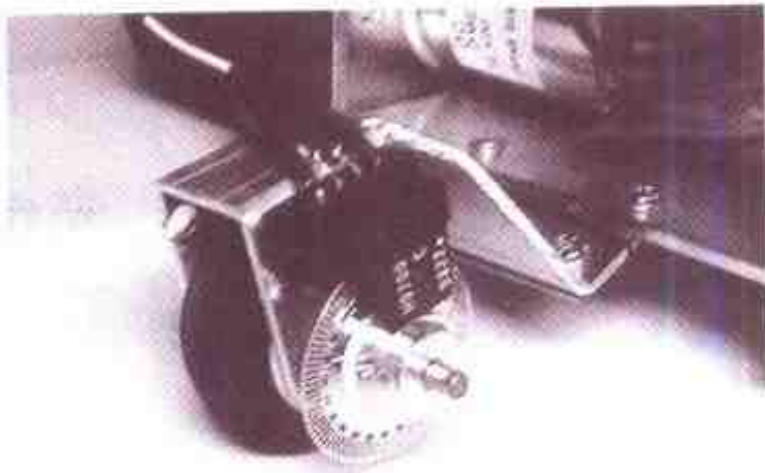
1. 日本微型计算机圣杯机器人(没有转向转换装置)的程序

其构成如照片 11. 10 所示,使用微型计算机和驱动电动机插件板、光传感器插件板及旋转编码器(参见照片 11.11)。



• 照片 11.10 从侧面看机器人

和微型计算机连接变为表 11. 1。根据这个连接实例程序工作。在程序中由于加入注释,希望参照执行。



• 照片 11.11 安装旋转编码器的情况

表 11.1 与微型计算机的连接

位	P7	PA	PB
0	光传感器	旋转编码器 A 相	—
1	光传感器	旋转编码器 B 相	—
2	光传感器	右电动机 PWM	—
3	光传感器	右电动机 ON-OFF	—
4	光传感器	左电动机 PWM	—
5	光传感器	左电动机 ON-OFF	—
6	光传感器	LED	—
7	光传感器	按钮开关	—

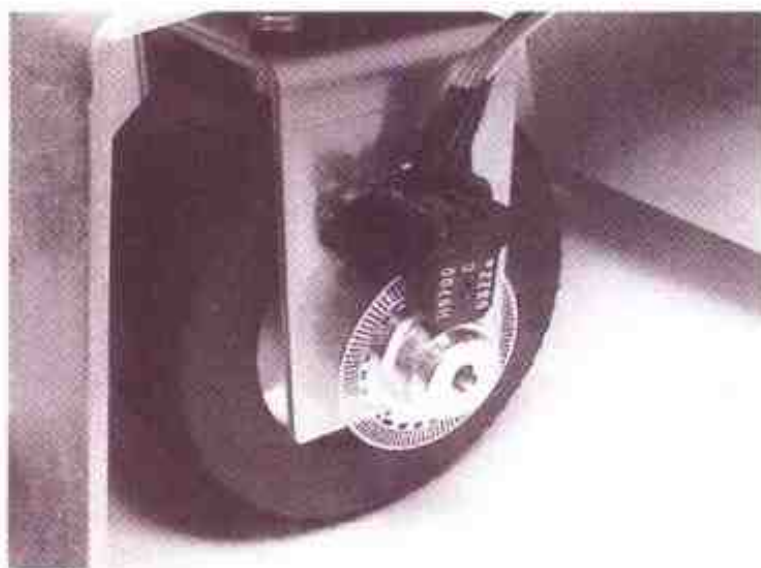
特别对控制担心点是使用分立式的 8 个传感器的地方,这里为了进行 PID 控制,将 8 个光传感器输出一次转换成模拟值进行控制。

2. 微型鼠大会机器人竞赛程序

在微型鼠大会上,进行机器人竞赛也可以用上述的机器人参加。连接方法完全相同,而必需变更程序。

3. 日本微型计算机圣杯机器人(带有转向转换装置的机器人)程序

对构成上述机器人补加转向转换装置用的伺服机构。另外,旋转编码器应该安装在前轮的车轮上,如照片 11.12 所示。因而对上述机器人轮胎直径不同,另外直角转向的转数定时根据车长的不同变更。



• 照片 11.12 安装旋转编码器的情况

与微型计算机的接线变为表 11.2。仅仅是增加到转向转换装置用的 PWM 端口连接。

表 11.2 与微型计算机的连接

位	P7	PA	PB
0	光传感器	旋转编码器 A 相	转向转换装置 PWM
1	光传感器	旋转编码器 B 相	—
2	光传感器	右电动机 PWM	—
3	光传感器	右电动机 ON-OFF	—
4	光传感器	左电动机 PWM	—
5	光传感器	左电动机 ON-OFF	—
6	光传感器	LED	—
7	光传感器	按钮开关	—

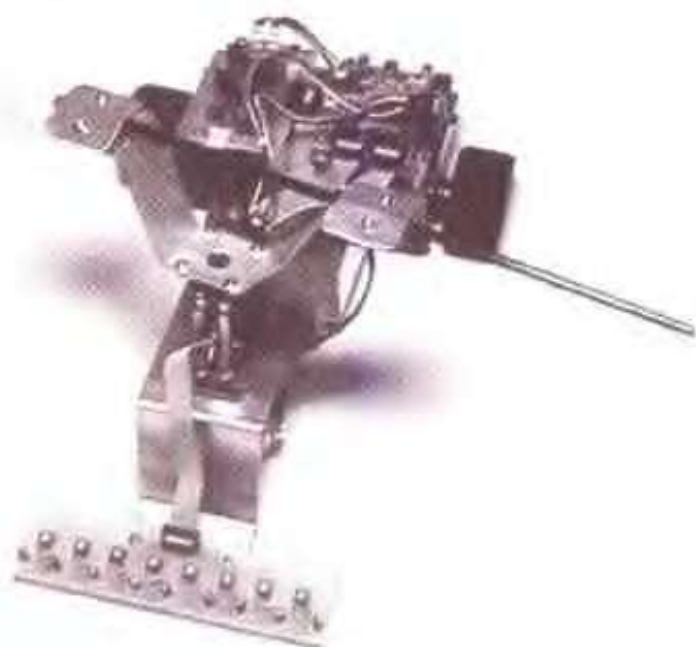
4. 三叉戟机器人的程序

对于上面带有转向转换装置的机器人,进一步地增加长枪部件(三叉戟)用的无线遥控用伺服机构。和微型计算机的连接是表 11.3。

表 11.3 与微型计算机的连接

位	P7	PA	PB
0	光传感器	旋转编码器 A 相	转向转换装置 PWM
1	光传感器	旋转编码器 B 相	—
2	光传感器	右电动机 PWM	长枪部件 PWM
3	光传感器	右电动机 ON-OFF	—
4	光传感器	左电动机 PWM	—
5	光传感器	左电动机 ON-OFF	—
6	光传感器	LED	—
7	光传感器	按钮开关	—

程序的要点就是一边在直线上追踪,一边向前行走。对直线部分尽可能减少控制量,使其稳定行走。为此,稳定的长枪部件可以撞击目标(参见照片 11.13 所示)。



• 照片 11.13 三叉戟机器人

chapter 12

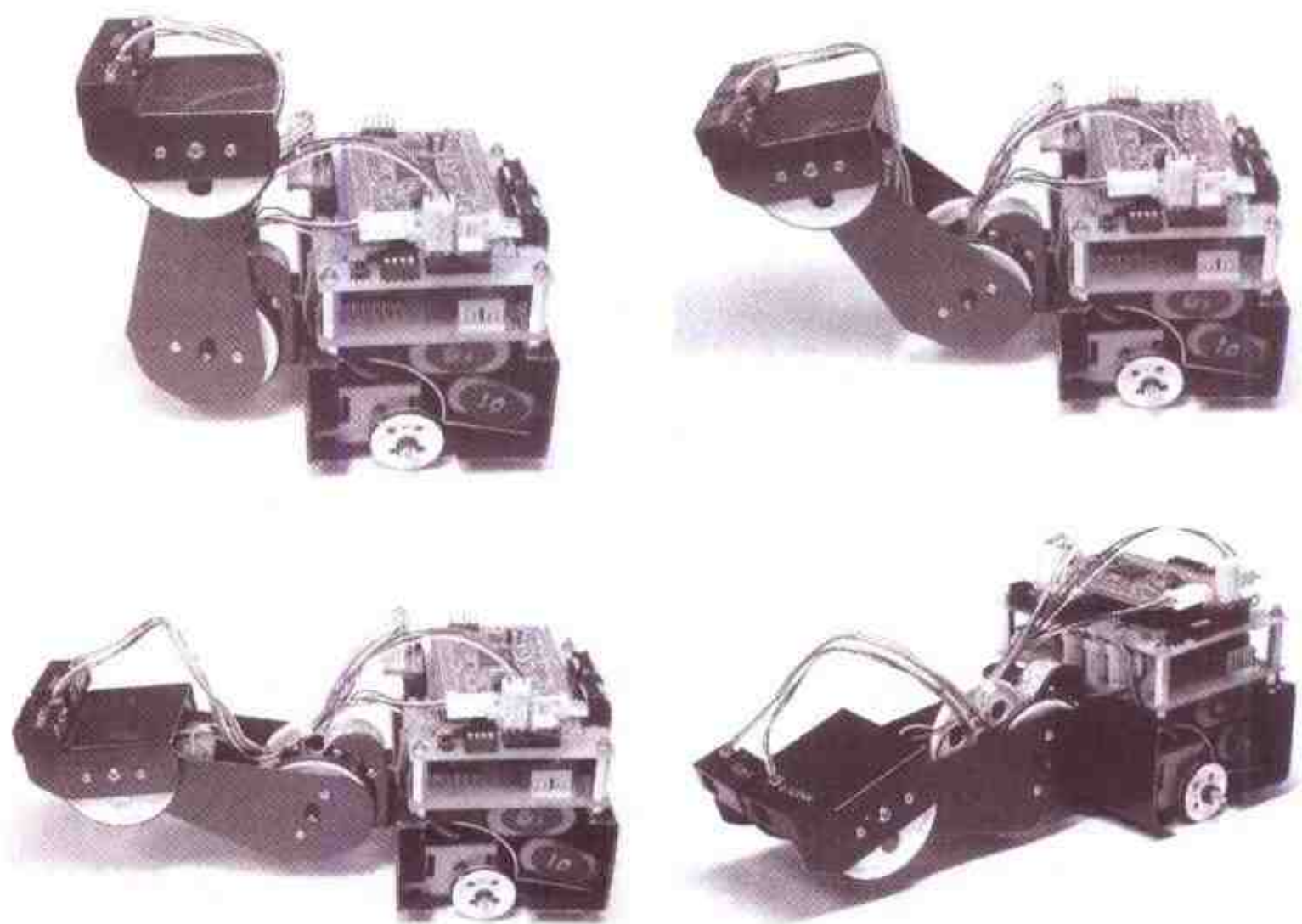
制作小型机器人

前面介绍了各种技术问题，以及能完成各种各样动作的小型机器人的制作。现在，试制作小型机器人。CPU或驱动电动机电路使用的部件和过去使用的相同。电动机使用MABVCHI电动机或在秋叶原商店可以买到的微型四轮驱动电动机、或是在录音机或随身听上使用的电动机等等，什么都可以。也就是说小型机器人无论在什么样的电动机驱动下都可以动作。

12.1 小型机器人概述

如照片 12.1 所示具有头和顶部,对于脚使用 2 个车轮,把前后、左右、头的上下和顶部的上下作为 4 轴机器人。使其成为步行型机器人,希望安装上脚,于是轴数变多,增加 CPU 的负担,所以本书中用轮胎移动机器人。

为了变成步行型机器人,最好是使用高性能的 CPU。另外,为了控制电动机还要有专用的 IC 模块等,而这里必需补加电路。为此,步行型机器人希望作为将来的课题。



• 照片 12.1 小型机器人

传感器为 2 个光传感器,来确定头和顶部运动在什么程度上是可能的。这样,随着不断的摸索,察觉到还可以进行各种各样的动作。使头上下

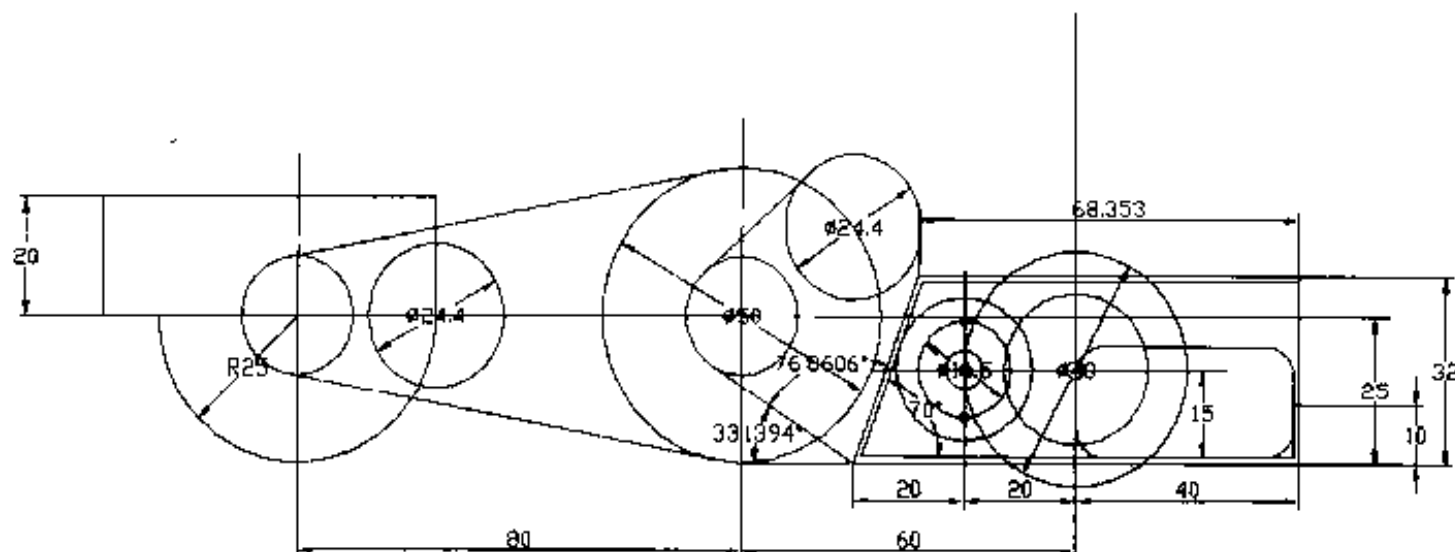
左右运动,应该可以知道对手的大小。对手大的时候避开,对手小的时候攻击,可以进行连续地动作等。另外,在头和顶部安装检测位置的传感器。通

过判断对手偏离的大小或变化的速度,可以得知是能触摸顶部,还是能进行竞赛。

仅有 2 个光传感器,而组合成头和顶部的运动时,可获得范围广泛的信息,可以形成各种各样的动作。制作这种机器人看看,以便有一个初步印象。

如果将传感器安装在相扑机器人上,即使仅是 ON-OFF 光传感器,通过扫描传感器,可以计算到对手的距离或也可以知道对手的速度。

仅仅看照片很难理解它的大小,所以图 12.1 中画出了小型机器人的外形尺寸。

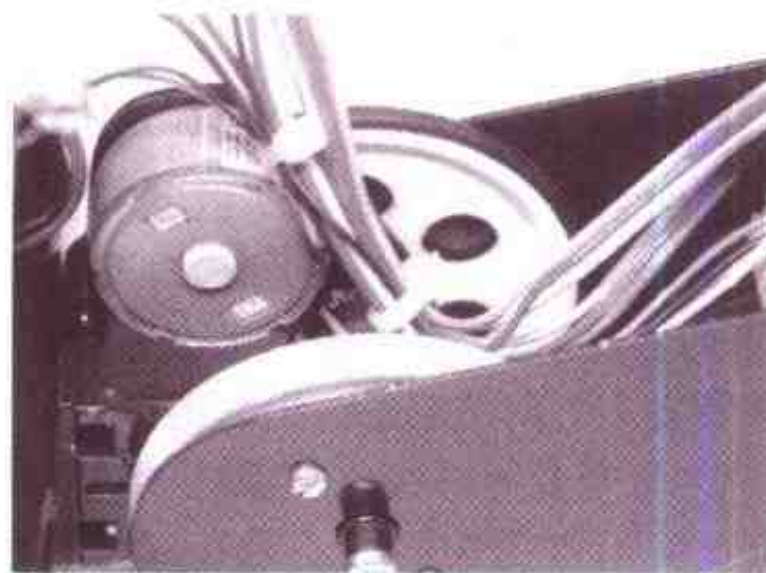


• 图 12.1 小型机器人的外形尺寸

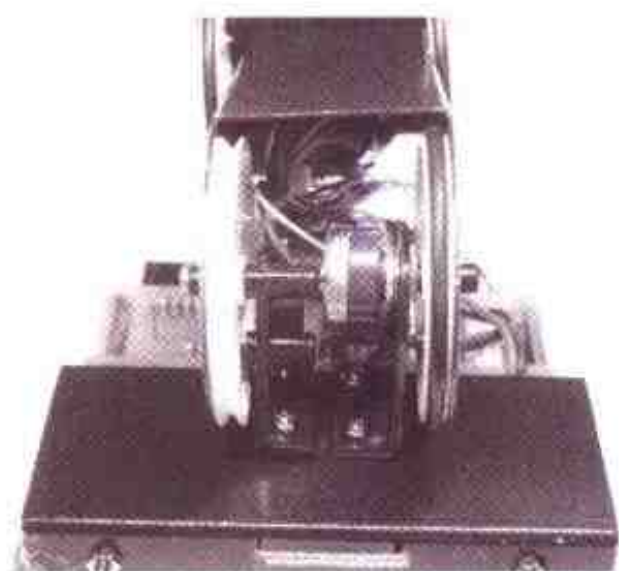
12.2 小型机器人的机械结构

12.2.1 头和顶部的机械结构

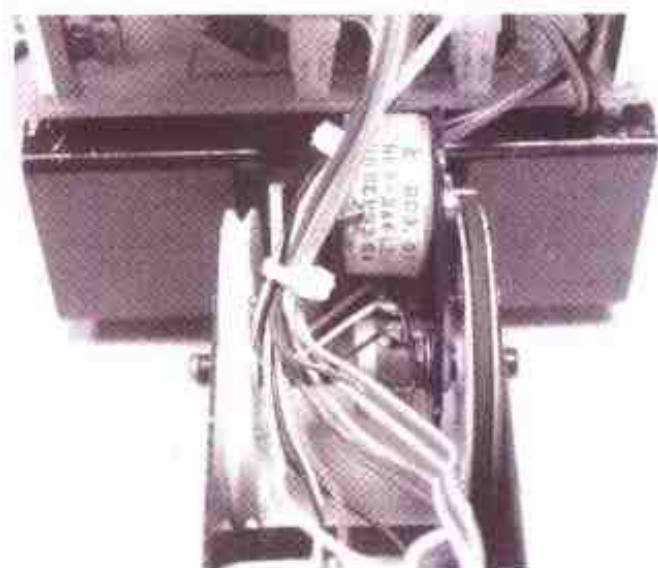
头和顶部的机械结构如照片 12.2,照片 12.3,照片 12.4 所示。将皮带轮安装在电位器上,然后,将电动机轴直接安装在皮带轮上,压紧 O 型环,使皮带轮旋转。对齿轮驱动系统驱动声音大,与其说是小型机器人,不如说是完全变成白铁皮玩具的感觉。为了使机器人没有噪音,采用 O 型环驱动。



• 照片 12.2 驱动头的部分(从侧面看到的地方)



• 照片 12.3 驱动头的部分(从下面看到的地方)



• 照片 12.4 驱动头的部分(从上面看到的地方)

皮带轮使用田宫模型的皮带轮组(参见照片 12.5 所示)。另外,在皮带轮部件组合上也装有电动机(参见照片 12.6 所示),所以使用它很方便。对于这些组合也包括塑料轴套等。如果买进 2 个皮带轮部件组合的零件备齐,为外表变得好看,在电动机不驱动侧也使用皮带轮即可。另外,O 型环在东急手工店或星期日木工商店可以买到。



• 照片 12.5 皮带轮组



• 照片 12.6 皮带轮部件组合(包括电动机)

为了使头转动,电动机使用在桌子中间闲置的部件,然而它应该是 MABUCHI 电动机会社制作的 RF-300C 电动机,额定电压为 5V 左右。

田宫模型网址:

<http://www.tamiya.com/>

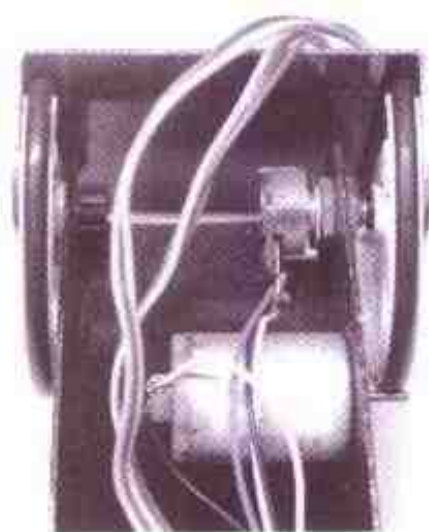
电位器轴使用 3mm 的部件即可。对装入皮带轮部件组合的塑料轴套有各种各样的尺寸,如照片 12.7 所示,

它是紧紧贴住轴的部件。将这个电位器用 1mm 的铝板固定在制作好的机器人的头部上,安装上直径 50mm 的皮带轮。特别详细的设计图没有表示,然而我认为是容易理解的。



• 照片 12.7 电位器和轴套

如照片 12.8 所示,顶部的部分完全和头的部分结构相同,皮带轮也使用相同直径 50mm 的部件。这里电动机使用稍微小一些的部件,这将点条式打印机卸开就可以得到。如果你步行在秋叶原,各种各样的电动机可以



• 照片 12.8 头和顶部的位置传感器、电位器

便宜地买到手。电动机额定电压使用 12 只镍镉蓄电池,所以使用 12V 左右的机器人是最好的,而用 PWM 电源,如果能进行限制电流,即使是 1.5V~3V 左右的电动机,也可以使用。但是,为了控制电动机,对其软件也必需充分注意。



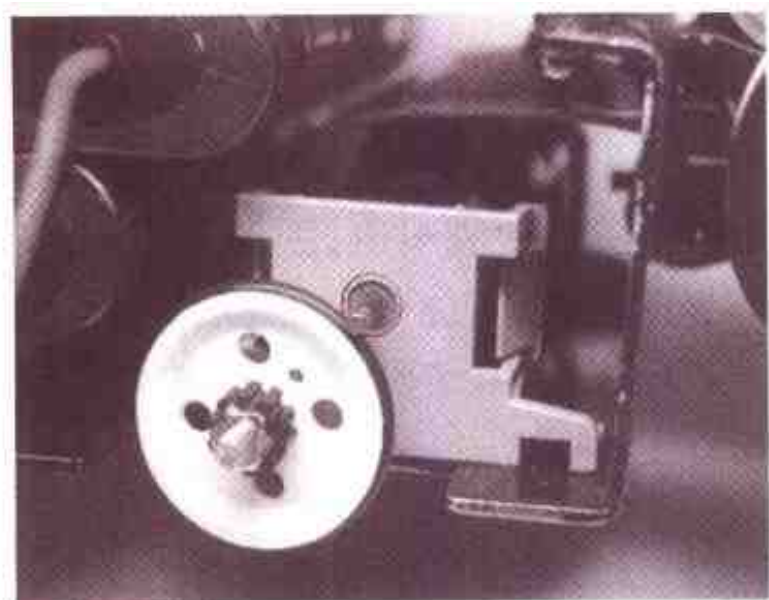
12.2.2 脚周围

脚周围也和头或顶部驱动部分相同,在皮带轮上装入 O 型环,将电动机轴压紧作为驱动方式。照片 12.9 是 O 型环、轴套和皮带轮。将 20mm 的 O 型环安装在皮带轮上使用。

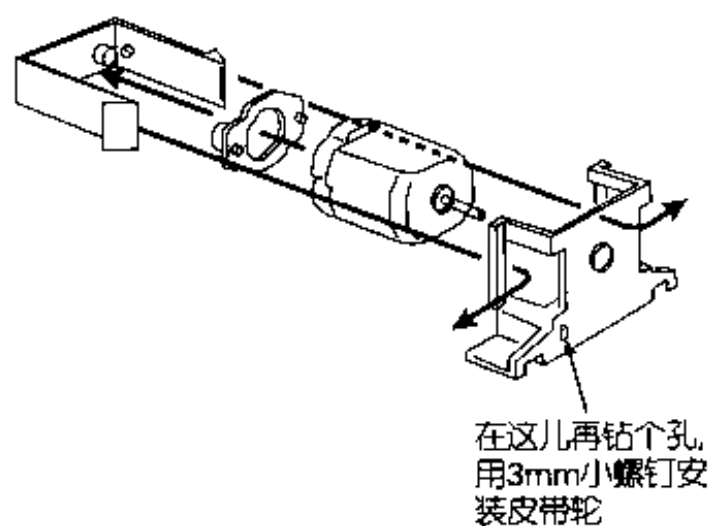


• 照片 12.9 O 型环、轴套和皮带轮

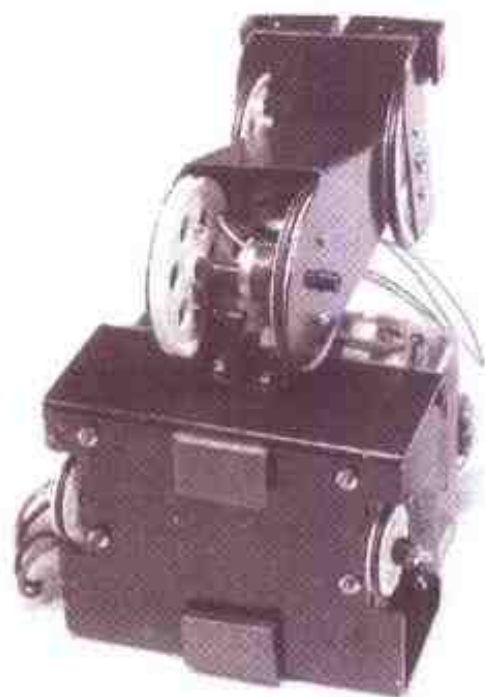
对脚周围的电动机使用 MABUCHI 电动机 FA-130,这就是装入皮带轮部件组合。如图 12.2 在装入皮带轮部件组合电动机的座上再钻 3mm 的孔,这里用 3mm 小螺钉,将皮带轮通过轴套如照片 12.10 安装即可。电动机座安装在机器人底盘上。如照片 12.11 用 2 个小螺钉固定。另外,在机器人底部的前后贴上家具导轨板。



• 照片 12.10 电动机和皮带轮



• 图 12.2 电动机和皮带轮的安装



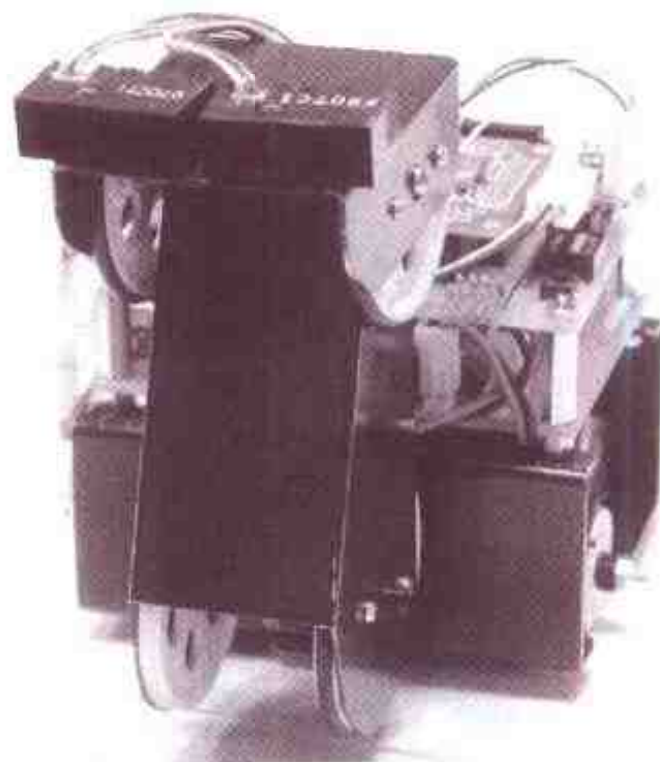
• 照片 12.11 从里面看到机器人的地方

12.2.3 小型机器人的眼睛

对于小型机器人的眼睛,使用 2 个最佳技术株式会社的 800mm 光传感器(参见照片 12.12),通过它检测障碍物。安装如照片 12.13 所示,连接在顶部的上面。光传感器安装在法兰部分有剩余,所以如照片 12.14 所示将其切掉一部分。

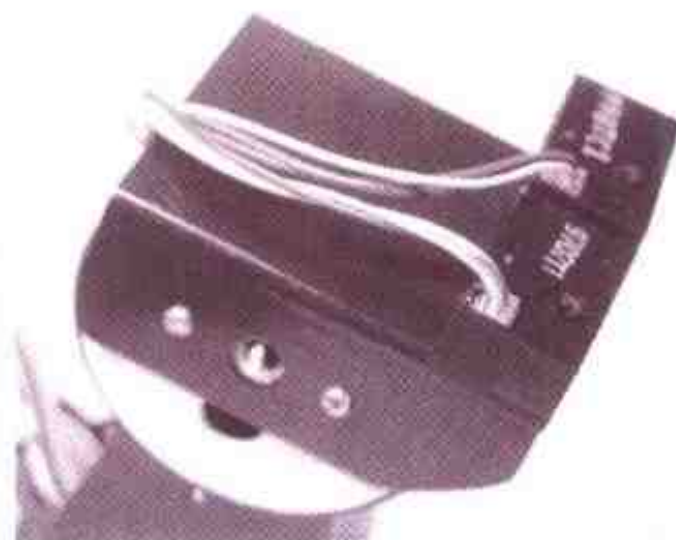


• 照片 12.12 反射型光传感器(BTE 023)



• 照片 12.13 从前面看到机器人的地方

最佳技术株式会社网址:
<http://www.besttechnology.co.jp/>



• 照片 12.14 安装光传感器的情况

12.3 小型机器人的回路

传感器及电动机与 CPU 端口的连接如表 12.1。

作为眼睛使用光传感器,输出 ON-

OFF 信号,这里使用光传感器变成扩充检测器输出,所以信号用 $1k\Omega$ 的电阻,电压提高到 5V。

表 12.1 CPU 端口和传感器、电动机的连接

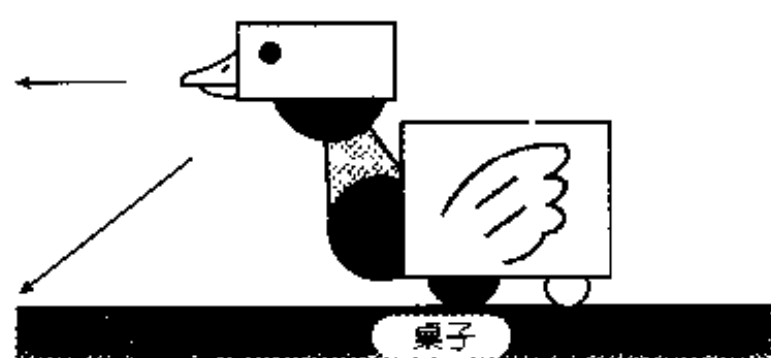
位	设定 P1	P2	P6	PA	PB	P7
0	右眼光传感器	LED 位 0	数字开关位 0	—	顶部电动机 PWM	顶部电位器
1	左眼光传感器	LED 位 1	数字开关位 1	—	顶部电动机 ON-OFF	头部电位器
2	—	LED 位 2	数字开关位 2	右电动机 PWM	头电动机 PWM	—
3	—	LED 位 3	数字开关位 3	右电动机 ON-OFF	头电动机 ON-OFF	—
4	—	LED 位 4	按钮 SW	左电动机 PWM	—	—
5	—	LED 位 5	—	左电动机 ON-OFF	—	—
6	—	LED 位 6	—	—	—	—
7	—	LED 位 7	—	—	—	—

12.4 小型机器人动作实例

小型机器人如何动作,是值得大家思考和探索的。下面简单地介绍一下动作的实例,图 12.3 为在桌子上行走的方法。首先,将头低下时,知道有

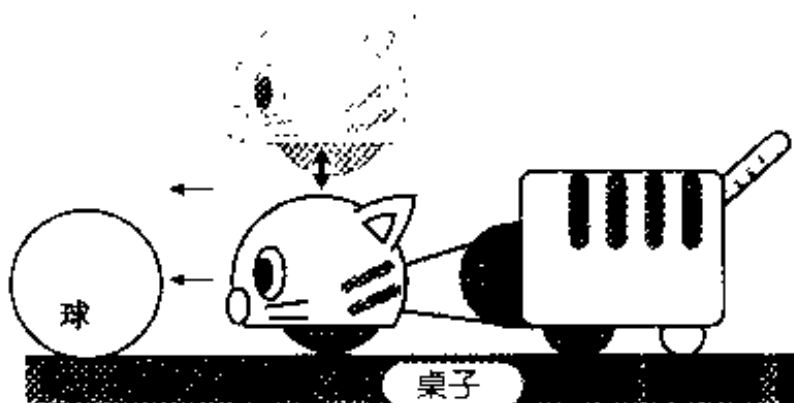
没有桌子,将头抬起来时,知道是否有障碍物。没有桌子的情况,沿桌子的方向返回,有障碍物的情况避开它。如果有这样的程序设计,机器人就不

会从桌子上掉下,并且一定避开障碍物步行返回。



• 图 12.3 在桌子上行走的小型机器人

图 12.4 是追赶球的动作。这里使机器人头上、下运动时认识球,在下侧时光传感器(眼睛)反应,在上面时为没有反应的条件。因此,如果和球是同样高度的物体,都认为是球,这里不必做太深入的考虑。



• 图 12.4 检测追赶球的动作

为此,需要设计一种可以判断在什么条件下,进行什么动作的程序。如一旦发现球,应该进入追赶的动作,追逐球和在相扑机器人中追赶对手是同样的。

为了丰富小型机器人,也期待大家给予更多的建议。

12.5 小型机器人的实例程序

下面介绍小型机器人的实例程序。由于尚未得到公证,所以使用时不负相应的责任,只是将它作为参考程序装入引起大家的兴趣,希望大家完成这个有意思的小型机器人。

“在机器人中加入电源时,头和顶部摇晃,使摇晃的头动作”、“不做游戏时进入睡眠状态”、“传感器虽有反应,由于睡眠不起作用,暂时不表示”、“发现球就追赶”等等,试进行上述这些动

作的程序设计。

程序设计后对自己可以欣赏,然而对初次见到的人不加以说明时,不容易得到理解,稍微有点失落感觉。

必须注意地是使用电动机额定电压降低机器人的情况。这时,充分增加 PWM 频率,降低占空比使用即可。这里由于机器人使用的电动机不同,所以将占空比下降到电动机不发热状态使用即可。

内 容 简 介

本书是引进日本 OHM 社版权,翻译出版的“机器人竞技系列”之一。该系列共六种,分别为《机器人竞赛指南》、《机器人制作宝典》、《机器人组装大全》、《自律型机器人制作》、《有视觉机器人制作》及《机器人 C 语言机电一体化接口》。

本书介绍了自律型机器人的种类,制作机器人所必须的电子元器件及其相应电路,H8 微型机结构的插件板、机器人感觉传感器及手、脚结构,并介绍了相扑机器人、直线追踪型机器人及小型机器人的制作方法。

本书可作为大、中、小学生参加竞技机器人大赛或制作竞技机器人的参考手册,也可供广大青少年竞技机器人爱好者阅读参考。

本书著作权和专有出版权受到《中华人民共和国著作权法》的保护。凡对本书的一部分或全部进行转载、或用复印机进行复制或在其场合引用以及录入电子设备等行为,均属侵害著作权,构成违法。

本书如需复制、引用、转载、改编时,必须得到版权所有者的许可。

如有任何疑问请与以下部门联系。联系时请尽量使用信函或传真形式。

科学出版社总编部 电话:010-64034529 传真:010-64019810

邮政编码:100717 地址:北京市东黄城根北街 16 号

<http://www.sciencep.com>

北京东方科龙图文有限公司 电话:010-82843276 010-82843277 传真:010-82842304

邮政编码:100029 地址:北京市朝阳区华严北里 11 号楼 3 层

<http://www.okbook.com.cn>

[G e n e r a l I n f o r m a t i o n]

书名 = 自律型机器人制作

作者 =

页数 = 1 8 6

S S 号 = 1 0 8 4 2 3 0 1

出版日期 =

封面
书名
版权
前言
目录
正文
封底